

Date NOT very good!

VVV

B

DOCUMENT-IDENTIFIER: US 6138161 A

TITLE: Method and system for maintaining reserve command relationships in a fibre channel network

BSPR:

More specifically, the present invention provides a method and system for maintaining a unique reserve command relationship between an initiator and a target device in a Fibre Channel network across network address changes following a break in communication. The method includes the step of maintaining in the initiator a target triplet table comprising a unique target triplet of data for the target device if the initiator is in communication with the target device. An initiator and a target device can be in communication if they are either in a unique reserve command relationship or if they have I/O transmissions in progress. The target triplet of data comprises a target device network address, a target device node name, and a target device port name. The target device may represent a device such as a SCSI router, for example, the Crossroads Systems, Inc. Model 4100. The Fibre Channel network may be a Fibre Channel arbitrated loop or switch network or other network topology.

CLPR:

9. The method of claim 8, wherein each of the plurality of initiators is in communication with at least one of the plurality of target devices.

CLPR:

10. The method of claim 8, wherein each of the plurality of target devices is in communication with at least one of the plurality of initiators.

CLPR:

25. The system of claim 24, wherein each of the plurality of initiators is in communication with at least one of the plurality of target devices.

CLPR:

26. The system of claim 24, wherein each of the plurality of target devices is in communication with at least one of the plurality of initiators.

servers. This latter embodiment is useful when, for example, **servers** become disabled or off-line.

DETD . . . the current invention in which resource rebalancing processes are set forth. Resource rebalancing includes re-mapping of pathways between nodes, e.g. **servers**, and resources, e.g. volumes/file systems. Load rebalancing allows the network to reconfigure itself as components come on-line/off-line, as components fail, . . .

DETD In the embodiment shown in FIG. 1C, memory resources 118A-B, **servers** 104C-106C, and normal **clients** 100A are shown, Memory resource 118A includes configuration database 120A1-D1. The cluster configuration database includes a clustered node database, a.

. layout of the resource. Memory resource 118B includes a plurality of file systems 122B1-3 and associated directory and access tables.

Server 104C includes processes 104PC, while **server** 106C includes processes 106PC. In the example shown, **server** 106C has twice the processing capability of **server** 104C.

DETD **Clients** 100A are connected via a network 108 to each of **servers** 104C-106C. Each of **servers** 104C-106C is connected to both memory resources 118A-B, via private network 112. In operation at time t=0, **server** 104C alone is operational. Processes 104PC cause **server** 104C to accept and process requests for any of file systems 122A1-3, and 122B1-3 on memory resources 118A-B. At time t=0, **server** 104C is shown accessing file systems 122A2-3, via paths 90A, file system 122A1, via path 90B, and file systems 122B1-B3, via paths 90C. At time t=1, **servers** 106C and 104C are operational. When **server** 106C comes on-line, resident processes 106PC seize control of the configuration database 120A1-D1 by placing a lock in the lock portion 120-D1 of the database. While this lock is in place, any other **server** attempting to rebalance the resources will see that rebalancing is taking place by another **server** when it fails to obtain the lock.

Server 106C thus becomes the temporary master of the resource rebalancing process.

DETD . . . configuration database records for all volumes and active nodes

to rebalance the system. Rebalancing the system takes into account preferred resource-**server** affiliations, expected volume traffic, relative **server** processing capability, and group priority and domain matches, all of which are contained in

configuration

database 120A1-B1. Optimal re-mapping between the existing **servers** 104C-106C and the available memory resources 118A-B is accomplished by processes 106PC. These results are replicated to each **server's** copy of the dynamic RAM resident configuration database 120A2-B2. The results are published and received by processes 104PC, on **server** 104C, and the lock 120D1 is removed. Subsequent to the load rebalancing, **server** 106C takes on responsibility for handling, via path 92B, I/O requests for file systems 122B1-B3. Further administrative access to these file systems, via paths 90C, from **server** 104C ceases. An additional path 92A, between **server** 106C and file system 122A1, is initiated and the path 90B, between that same file system and **server** 104C, is terminated. Thus, after resource rebalancing, **server** 106C handles I/O requests for four out of the six file systems, namely

122A1,

122B1-B3, while **server** 104C handles only file systems 122A2-3. Several embodiments of the load rebalancing embodiment just discussed will be set forth in. . .

DETD . . . and variations thereof can be practiced individually, or in combination, without departing from the teachings of this invention.

For example, **client** load rebalancing and distributed I/O can be combined. **Client** load rebalancing and resource rebalancing can be combined. Distributed I/O and resource rebalancing can be combined. **client** load rebalancing, distributed I/O, and resource rebalancing can be combined.

DETD FIG. 2A shows the software modules present on **server** 104 for enabling **client** load balancing, distributed I/O, and resource rebalancing embodiments of the current invention. FIG. 2A shows **server** 104 and memory resource 118. **Server** 104 includes a logical I/O unit 130 and a physical I/O unit 132. The logical I/O unit includes an internal . . . module 148, a command processing module 154, a disk reader module 150, a shared data metadata management module 152, a **server** configuration driver 156, a **resource management** module 158, a logical name driver module 160, and a metadata supplier module 162. The physical I/O unit 132 includes. . .

DETD . . . 140, the command receipt module 142, the shared data lock management module 144, the configuration database replicator module 148, the **resource management** module 158, the **server** configuration driver 156, the shared data metadata management module 152, the metadata supplier module 162, the disk reader module 150, and I/O store and forward 166. The **resource management** module 158 is connected to the resource publisher 146 and to the logical name driver module 160. The metadata supplier. . .

DETD . . . are received and queued up, either from internal I/O module 140, from the private network 112 (from a data transfer **server**), or from a normal or aware **client** on network 108. The I/O is thus tagged with the source type for future decision making.

DETD . . . resources on this node. It is the module that actually interacts with the network in order for normal and aware **clients** to figure out which resources are available on this node. The resource publisher 146 interacts with the **resource management** module 158 and logical name driver module 160 to obtain the actual information that should be published in the network. . .

DETD . . . namespace, and provides a path for the logical name driver

Updated 5 09/401,874

	L #	Hits	Search Text	DBs
1	L9	41686	5and 6	USPAT; US-PGPUB
2	L8	1	7 and 4	USPAT; US-PGPUB
3	L7	2	5 same 6	USPAT; US-PGPUB
4	L6	41168	(indicat\$4 or identify\$4) adj3 (fail\$4 or trouble\$ or problem\$ or error\$)	USPAT; US-PGPUB
5	L5	389	1 with 2	USPAT; US-PGPUB
6	L4	736343	router\$ or switch\$3 or bridge\$	USPAT; US-PGPUB
7	L3	102287	(plural\$4 or multipl\$4) adj2 (unit\$ or device\$ or system\$)	USPAT; US-PGPUB
8	L2	4216	target adj1 (device\$ or unit\$ or system\$)	USPAT; US-PGPUB
9	L12	19	10 and 4	USPAT; US-PGPUB
10	L11	0	10 and 4 and 6	USPAT; US-PGPUB
11	L10	23	5 same 3	USPAT; US-PGPUB
12	L1	602528	communicat\$4	USPAT; US-PGPUB

	Time Stamp
1	2001/07/18 16:26
2	2001/07/18 16:23
3	2001/07/18 16:26
4	2001/07/18 16:21
5	2001/07/18 16:19
6	2001/07/18 16:19
7	2001/07/18 16:19
8	2001/07/18 16:12
9	2001/07/18 16:27
10	2001/07/18 16:27
11	2001/07/18 16:27
12	2001/07/18 16:07

DOCUMENT-IDENTIFIER: US 5790775 A

TITLE: Host transparent storage controller failover/failback of SCSI targets and associated units

BSPR:

However, the direction taken by Idleman requires a multi-level storage controller implementation. Each controller in the dual-redundant pair includes a two-level hierarchy of controllers. When the first level or host-interface controller of the first controller detects the failure of the second level or device interface controller of the second controller, it re-configures the data path such that the data is directed to the functioning second level controller of the second controller. In conjunction, a switching circuit re-configures the controller-device interconnections, thereby permitting the host to access the storage devices originally connected to the failed second level controller through the operating second level controller of the second controller. Thus, the presence of the first level controllers serves to isolate the host computer from the failover operation, but this isolation is obtained at added controller cost and complexity.

DEPR:

Returning to FIG. 1, physical storage media 32, which is comprised of SCSI I/O devices 34, is interconnected to each controller 14 on the device side SCSI bus 26. Each is identified by a SCSI bus address, physically implemented in the device by switches on the device or by suitable jumper connections programming default bus address information in the form of binary address for the device. The SCSI I/O devices 34 in the preferred embodiment shown in FIG. 1 are disk drives, but the principles of the present invention may be extended to systems utilizing other SCSI compatible peripherals and I/O devices.

DEPR:

A preferred implementation for the storage controller 14 (from FIG. 1) is illustrated by the block diagram shown in FIG. 4. The storage controller 14 bridges the host side SCSI bus 16 via the SCSI host port 18 to one or more of the device side SCSI buses 26 attached to corresponding SCSI device ports 28. Referring to FIG. 4, the storage controller 14 further comprises a policy processor 42, which controls all but the low-level device and host port operations. Sharing a native bus 44 used by the policy processor are a nonvolatile memory 46, diagnostics and control registers 48, a maintenance terminal port 50 and dual controller or communications port 52. The nonvolatile memory 46 holds controller firmware 54 as well as parameter information 56 entered by the user and by the controller software. Typically, the portions of the nonvolatile memory storing these components are physically implemented in separate memory devices. Part of the firmware (i.e., boot diagnostics) executes from the nonvolatile memory, but the majority of the diagnostics, and all of the functional code and utilities are actually run by the policy processor from a shared memory 58. The shared memory 58 includes buffer memory and memory control support logic. The firmware is copied from the nonvolatile memory 46 to the shared memory 58 by the boot diagnostics each time the controller boots.

DEPR:

In the preferred implementation, the transparent failover process simulates a POWER FAIL situation. The failover action appears to the host CPU as a power failure, in which there is normally a complete controller system reinitialization. When it appears to the host CPU that the reinitialization is complete (i.e., the failed controller's ID or IDs appear to be back "online"), the host CPU resumes communications with the target IDs of the failed controller 112. All the while, the surviving controller is still running with its own ID or IDs to the host CPU. In response to host CPU communications to an ID of the failed controller, the surviving controller sends the host CPU a check condition status 114. The check condition status indicates to the host CPU that a problem or exception condition has occurred. The host CPU then sends a response to check condition status by sending request/sense command requesting information 116. Sense data describing a power on and reset event is subsequently sent by the surviving controller 118. With reference to FIG. 7B, the host CPU now believes that the target ID lost power momentarily and is forced to go back through a SCSI initialization sequence 120. Once the initialization has completed, the host CPU is ready to re-issue any outstanding SCSI commands to a given LUN or unit associated with the failed controller. First, however, the host CPU tests the readiness of the unit 122. If the unit is not ready, the host CPU needs to issue a start command to make the unit ready on the SCSI bus 124. The host CPU also wants to ensure that it is still communicating with the same target device that it was communicating with prior to the "power failure". At minimum, it will poll the target device for ID inquiry data describing that device 126. Finally, the host CPU reissues any outstanding commands issued to that target ID 128. The surviving controller is now running with the IDs of both the surviving controller and the failed controller 130. Therefore, "normal" operations resume until such time as a failback operation occurs.

DOCUMENT-IDENTIFIER: US 6182182 B1

TITLE: Intelligent input/output target device communication and exception handling

DEPR:

In most circumstances, proper communication between the block storage OSM 202, the target device 204 and ultimately, with the desired peripheral device, will operate very lean on host processor overhead. However, in those cases where an error occurs, the exception OSM driver 206 will swiftly move in to determine what the error may have been, attempt to repair the error if possible, and then provide the proper I.sub.2 O reply to the block storage OSM 202. In some cases, the I.sub.2 O exception reply provided by the exception OSM driver 206 will be a reply that indicates completion with error or retry occurred. In other cases, the reply will simply indicate that an error occurred and that the error could not be remedied.

DOCUMENT-IDENTIFIER: US 6260093 B1

TITLE: Method and apparatus for arbitrating access to multiple buses in a data processing system

ABPL:

A method and apparatus in a data processing system for multiple bus arbitration, wherein the data processing system includes a first bus connected to a second bus by a bridge. In response to receiving a request for a target device from a master device connected to a first bus, a determination is made as to whether the target device is connected to the first bus. The bridge is selected in response to determining that the target device is located on the second bus. The bridge initiates a request for the second bus in response to the selection of the bridge. The first bus and the second bus are connected to

each other by the bridge in response to the bridge receiving a grant to the second bus, wherein the master device transfers data between the master device and the target device across the bridge. In response to the bridge being selected from a master device on both the first bus and the second bus, the bridge signals one master device to retract or withdraw the selection of the bridge, allowing the other master device to complete a data transfer.

BSPR:

In data processing systems containing multiple buses and multiple master devices, in which the master devices communicate with devices on other buses, a

system of arbitration on multiple buses is required for high performance and reliability of avoiding deadlock situations in which master devices on different buses make requests for target devices or resources on opposite sides

of the buses. Presently available arbitration systems include a complex hierarchical arbitration system that determines all possible deadlock situations up front in designing the system. In such an arbitration system, all of the deadlock situations are designed into a top level arbiter. This top

level arbiter, directed lower level arbiters on the bus level to avoid deadlock. The drawback of such an arbitration system is that is a potential deadlock condition was missed, the chip could lock up. Therefore, an improved method and apparatus for bus arbitration that avoids deadlock situations for multiple bus data processing systems is desirable.

BSPR:

The present invention provides a method and apparatus in a data processing system for multiple bus arbitration, wherein the data processing system includes a first bus connected to a second bus by a bridge. In response to receiving a request for a target device from a master device connected to a first bus, a determination is made as to whether the target device is connected

to the first bus. The bridge is selected in response to determining that the target device is located on the second bus. The bridge initiates a request for

the second bus in response to the selection of the bridge. The first bus and the second bus are connected to each other by the bridge in response to the bridge receiving a grant to the second bus, wherein the master device transfers

data between the master device and the target device across the bridge. In

response to the bridge being selected from a master device on both the first bus and the second bus, the bridge signals one master device to retract or withdraw the selection of the bridge, allowing the other master device to complete a data transfer.

DRPR:

FIG. 4 is a flowchart of a process employed by a bridge during arbitration for access to a bus in accordance with a preferred embodiment of the present invention; and

DEPR:

With reference now to the figures, and in particular with reference to FIG. 1, a block diagram of a data processing system 100 in which the present invention may be implemented is illustrated. In particular, the present invention implements within data processing system 100 a single level arbiter and bridge system for arbiting requests across buses. Data processing system 100 employs an advanced system bus (ASB), which is part of the Advanced Microcontroller Bus

Architecture (AMBA) from Advanced RISC Machines, Ltd.(ARM). The bus is described in the AMBA specification, which is available from ARM located in Cambridge, England. Although the depicted example employs an ASB of the AMBA specification, other bus architectures used for system on a chip buses may be used as well as other bus architectures such as peripheral component interconnect (PCI) local bus, Micro Channel, and ISA may be used. In the depicted example, data processing system 100 includes bus 102, bus 104, bus 106, and bus 108. Bus 102 and bus 104 are connected to each other through bridge 110, while bus 104 and bus 106 are connected to each other through bridge 112. Bus 108 is connected to bus 104 by bridge 114. Master device 116, resource 118, arbiter 120, and decoder 122, are connected to bus 102. Arbiter 120 arbitrates access to bus 102, while decoder 122 decodes address placed onto bus 102. Resource 124 and resource 126 are connected to bus 104.

Additionally, arbiter 128 and decoder 130 are connected to bus 104. Arbiter 128 and decoder 130 provide arbitration and decoding functions for bus 104. Master device 132, master device 134, and resource 136 are connected to bus 106. Arbiter 138 and decoder 140 are connected to bus 106 and provide arbitration and decoding functions for bus 106. Bus 108 has a master device 142 and a resource 144 connected to it. Arbiter 146 and decoder 148 provide arbitration and decoding functions for bus 108.

DEPR:

Within data processing system 100, master devices located on each of the buses are able to concurrently access resources on their individual buses. When a master device, such as master device 116, wants to access a resource located on a different bus, such as resource 126 on bus 104, the transaction must cross a bridge, such as bridge 110. Decoder 122 selects bridge 110, which causes bridge 110 to arbitrate for bus 104. When an acknowledgement is received, the bridge, bridge 110, acts as a master device on a second bus, such as bus 104, and as a resource on the first bus, such as bus 102, in accordance with a preferred embodiment of the present invention. Thus, in addition to connecting the buses to each other to move data, bridge 110 also acts like a master device or target device on both buses in accordance with a preferred embodiment of the present invention.

DEPR:

The address from master device 116 on bus 102 is passed on to decoder 130 on bus 104 through bridge 110 connecting two buses. Depending on whether both buses, bus 102 and bus 104, use the same address map, bridge 110 may or may not perform address translation before passing the address to bus 104.

DEPR:

On bus 104, decoder 130 selects either a device or another bridge depending on the address placed on the bus by the master device. If the address is for a device on the second bus, bus 104, decoder 130 on bus 104 will select that device. If the address is for the device on the third bus, bus 106, decoder 130 would select bridge 112. If the target device were located on bus 108, decoder 130 would select bridge 114. Assuming that bridge 112 is selected, the process repeats with the address being passed to a third bus decoder, decoder 140. In accordance with a preferred embodiment of the present invention, the arbiter on each bus treats a bridge request like any other request from a master device. The arbiter on a bus may use any appropriate type of arbitration for the bus. The type of arbitration on each bus may be independent from the other buses.

DEPR:

In accordance with a preferred embodiment of the present invention, each bridge is able to recognize when it is selected by decoders on each side of the bridge at the same time. When such a situation occurs, the bridge issues a retry signal to one of the masters on one bus and processes the select from the master on the other bus. The master issued the retry signal removes its request from the bus. In this manner that bus is freed up so that the transaction on the other bus may be completed, thus, avoiding deadlock. Depending on the priorities of the devices in the data processing system, the bridge may alternate which side is issued a retry signal or always have one side issued a retry signal.

DEPR:

With reference next to FIG. 2, a block diagram of data flow in a data transaction between a master on a first bus and a target device on a second bus in a data processing system is depicted in accordance with a preferred embodiment of the present invention. Data processing system 200 includes bus one 202 connected to bus two 204 by bridge 206. Master device one 208 sends a request for bus one 202 to arbiter one 210. A grant is sent back to master device one 208. In the depicted example, decoder one 212 decodes an address placed on bus one 202. Based on the address, decoder one 212 selects bridge 206, which in turn requests bus two 204 from arbiter two 214. When a grant is received by bridge 206 from arbiter two 214, bridge 206 places the address on bus two 204. Decoder two 216 decodes the address and selects target two device 218. At this time, master device one 208 has access to target two device 218 to perform data transfer. A more detailed description of the process followed by these devices are described with reference to FIG. 3.

DEPR:

Turning next to FIG. 3, a flowchart illustrating transactions between a master on a bus that read/writes data to a target on a different bus is illustrated in

accordance with a preferred embodiment of the present invention. The process described in FIG. 3 is made with reference to the components illustrated in FIG. 2. The process begins with the master device requesting bus one (step 300). The master device sends a request to the arbiter on bus one. This arbiter can use any arbitration scheme, such as, for example, priority or round-robin. In this example, the master device only sends a request to the local bus, bus one. Next, arbiter one grants bus one to the master device (step 302). The grant of the bus by arbiter one to the master device is made with a grant signal. The master device then starts the transaction by placing the address of the destination or target device onto bus one (step 304). Decoder one on bus one decodes the address and selects the bridge (step 306). The decoder sees the address from the master device and recognizes that the target device is not on bus one. In accordance with a preferred embodiment of the present invention, decoder one selects the bridge. In the depicted example, the target is on bus two with the decoder selecting the bridge connecting bus one to bus two. If more than one bridge is connected to bus one, the decoder selects the correct bridge depending on where the target is located. The decoder selects the bridge based on the address placed onto the bus by the master device. In the depicted example, the target device is located on bus two. As a result, the bridge requests bus two from arbiter two (step 308). In other words, in step 308, the bridge sees the device selection from decoder one on bus one and generates a request to arbiter two on bus two. Arbiter two may use any type of arbitration scheme for its local bus, bus two. This arbitration scheme may be different from the one employed by arbiter one on bus one. For this transaction on the bus one side, the bridge acts as a target device on bus one, and for the transaction on the bus two side, the bridge acts as a master device on bus two.

DEPR:

At the same time as the bridge requests bus two, the bridge also may perform an address translation (step 310). This step is an optional step and is employed if bus one and bus two use different address maps. Also, concurrently with steps 308 and 310, the bridge waits the master device (step 312). From the time the bridge receives the device select signal from the decoder on bus one until the time the bridge connects bus one and bus two signals together, the bridge will cause the master device to wait. The process then proceeds when bus two is granted to the bridge (step 314). This grant occurs by the arbiter sending a grant signal to the bridge. Next, the bridge drives the address (or the translated address) onto bus two (step 316). Decoder two on bus two recognizes that the target device is located on bus two and issues a device select to the target device (step 318). At that time the bridge connects address, data, and control signals of bus one and bus two (step 320). This step occurs once the target device has been selected. The bridge connects the address, data, and control signals of bus one and bus two together so that the two buses will behave as one bus. The master device is no longer waited by the bridge and can now directly read and write data to the target device.

DEPR:

The master finishes the data transaction, removes the request for the bus, stops driving the address onto bus one, and performs other actions associated with the termination of the data transaction and the need for the bus. (step 322). In response, arbiter one removes its grant of the bus one to the master device by deasserting the grant signal to the master device (step 324) and decoder one, recognizing the end of the transaction, deselects the bridge by removing the device select to the bridge (step 326). In response to the

bridge

being deselected, the bridge removes its request for bus two and stops driving the address onto bus two and breaks all signal connection between bus one and bus two (step 328). As a result, arbiter two removes its grant of bus two to the bridge (step 330) and decoder two deselects the target device (step 332) with the process terminating thereafter.

DEPR:

In accordance with a preferred embodiment of the present invention, deadlock is

avoided by the bridge being able to recognize when it is selected by decoders on each side of the bridge at the same time. When this situation occurs, a bridge will issue a retry to one of the selected masters on the bus and process

the selection from the decoder on the other bus. Under this mechanism, the master that is told to retry its request, removes its request from the bus.

In this manner, the bus is freed up for the transaction on the other bus until the

transaction is complete, avoiding a deadlock. Depending on the priorities of the devices located in the data processing system, the bridge may alternate which side is issued a retry or always have one side issue a retry. This mechanism is described in more detail in FIG. 4 below.

DEPR:

Turning now to FIG. 4, a flowchart of a process employed by a bridge during arbitration for access to a bus is depicted in accordance with a preferred embodiment of the present invention. The process begins with the bridge in an idle state (step 400). In the idle state no transactions are crossing the bridge, and the two buses connected by the bridge are not connected to each other. A determination is made as to whether the bridge has been selected (step 402). This determination determines whether a decoder on one of the buses connected to the bridge has selected the bridge. If the bridge has not been selected, the bridge returns to the idle state in step 400. If the bridge

has been selected, the bridge determines whether it has been selected only by a decoder on one bus also referred to as being selected from "one side" or by decoders from both buses, also referred to as being selected from "both sides" (step 404). If a select is detected from both sides, it means that master devices from both buses are attempting to cross the bridge at the same time. In such a situation, the bridge issues a retry signal to one side (step 406) with the process then returning to step 404. Basically, one of the master devices is to be stopped from crossing the bridge and told to try its transaction at a later time. The bridge signals the master device on one of the buses that the bridge is busy and to retry at a later time. From the master's point of view, the bridge is the target device that it has addressed and that the target device has just told the master device that the target device is busy. Depending on the bus structure and the particular implementation, the bridge may be programmed to alternate which side is issued the retry signal or to always issue one side a retry signal.

DEPR:

If only one side has selected the bridge, the bridge initiates a bus request to the target bus (step 408). In the depicted example, the bus containing the master device is the first bus or the "master bus" and the bus containing the

target device is the second bus or the "target bus". At the same time, an optional address translation may be performed (step 410). The bridge performs this step if the two buses connected to the bridge are using different address maps. The bridge also waits the master at the same time (step 412). More specifically, the bridge signals the master on the first bus to wait. The purpose is to cause the master device to remain on the first bus without advancing the data transaction (i.e., do not increment the address). The wait signal issued to the master device is in effect telling the master device that the target device is slow.

DEPR:

From step 408, the bus monitors to determine whether a bus grant has been received (step 414). If a bus grant has been received, the address received from the master device is driven onto the target bus (step 418). If necessary,

this address may be a translated address generated from step 410. At this time

the bridge is acting like a master device on the target bus. The bridge is repeating the original master device's initiation of the transaction. This process is the reason that the bridge in step 412 waits the original master device. After driving the address onto the target bus, the bridge connects the

two buses, the bus containing the master device and the bus containing the target device (step 420). In step 420, the bridge stops waiting the master device and makes the connection between the master and target buses. The bridge provides a direct connection between the target device and the master device with no latency in the data transfer between the two devices.

DEPR:

A determination is made as to whether the device select has been removed from the bridge (step 422). If the select has not been removed, the process returns

to step 420. Otherwise, the bridge removes the request for the target bus and disconnects the master bus and the target bus from each other. The bridge then

returns to the idle state in step 400 to monitor for another device select.

DEPR:

With reference again to step 414, if the bridge has not received a bus grant, a

determination is made as to whether the bridge has received a device select from the target bus (step 426). If a device select has not been received, the process returns to step 414. Otherwise, the bridge issues a retry signal to one of the two master devices (step 428). Receiving a device select instead of

a bus grant means that a master device from the target bus side has been grant the bus instead of the bridge. Such a select of the bridge also means that a master device on the second bus is trying to cross the bridge to initiate a data transfer. The bridge must decide which master device is to continue the data transfer. The bridge could determine that the original master device on bus one is to continue the transaction and issue the retry signal onto the second bus. Alternatively, the bridge may determine that the new requesting master device on bus two should continue the transaction and issue the retry signal onto the first bus.

DEPR:

Next, a determination is made as to whether only a single device select remains

on the bridge (step 430). If two selects are still present, the process returns to step 428. Otherwise, a determination is made as to the master device selected for issuance of the retry signal was the original master device, the master device on the first bus (step 432). If the selected master for the retry signal is the original master device, the process removes the request from the second bus (step 434) and returns to steps 408, 410, and 412 as described above. In such a situation the first bridge sends a request to the first bus after removing its request from the second bus--the first bus becomes the "target bus" and the second bus becomes the "master bus". Otherwise, the process returns to step 414 as previously described.

DEPR:

The process followed by the bridge in FIG. 4 can be applied to a situation in which the target does not exist on the second bus, but on a third bus connected

to the second bus by a second bridge. In such a situation, the bridge drives the address onto the second bus with the decoder on the second bus selecting the second bridge connecting the second bus to a third bus on which the target device is located. The first bridge does not know that the target is not on the second bus. This process can be extended to any number of buses to the bus

on which the target device is located.

DEPR:

Turning now to FIG. 5, a flowchart of a process implemented in a decoder is depicted in accordance with a preferred embodiment of the present invention. The process begins by the decoder monitoring to determine whether an address is

valid on the bus (step 500). If an address is not present on the bus, the process returns to step 500. When an address is valid, the address is decoded (step 502). Each device in the data processing system is associated with an address or a range of addresses. A determination is made as to whether the decoded address is for a device located on the local bus (step 504). If the address is for a device on the local bus, the decoder then selects that device (step 506). Otherwise, the decoder selects a bridge associated with the decoded address (508). This bridge may be connected to a bus containing the device that is to be accessed or to a bus connected to a second bridge that is connected to a bus containing the target device. After selecting a device or a

bridge, the decoder determines whether the address is still valid on the local bus (step 510). If the address is still valid, the decoder continues to select

the selected device (step 512). Otherwise, the decoder deselects the selected device (step 514).

DEPR:

Thus, the present invention provides an improved method and apparatus for arbitrating access to devices on remote buses while avoiding deadlock situations occurring from master devices on two side of a bridge simultaneously

trying to cross the bridge to initiate a data transaction. The arbitration system of the present invention allows for multiple buses to be connected together. These buses may operate independently or they may operate together or a mixture of both. The arbitration system of the present invention resolves

all possible deadlock situations. The advantage is provided by processes implemented within the bridge that allows the bridge to act like a master or a target device. The bridge resolves all deadlock situations. As a result,

preplanning for deadlock conditions in the top level arbiter is not required. The deadlock is resolved by retracting one of the master devices so that only one master device selects a bridge. Additionally, decoders are designed to select a bridge when a decoded address on the local bus is for a device on a remote bus. When connecting a master device and a target device, no latency occurs in the data transaction after the bridge connects the buses together.

DEPR:

In addition, the distributed arbitration scheme of the present invention allows for different types of arbitration to be used on each bus. Also, different address schemes may be used on each bus with the bridge providing address translations when necessary. Also, the present invention allows for any number of target devices or master devices to be on a bus. A bus may contain all target devices or all master devices. Further, some devices may act as both target and master devices. Additionally, the present invention may support any number of buses. Crossing of multiple bridges from a master device to a target device in a single transaction is supported by the present invention.

CLPR:

1. A method in a data processing system for facilitating a data transfer between a master device and a target device, wherein the data processing system includes a first bus connected to a second bus by a bridge the method comprising:

CLPR:

13. The data processing system of claim 7, wherein the bridge connecting the first bus to the second bus is a first bridge, the master device connected to the first bus is a first master device, and the target device connected to the second bus is a first target device, the data processing system further comprising:

CLPR:

14. The data processing system of claim 13, wherein the second master device is the first bridge.

CLPR:

15. A bridge comprising:

CLPR:

16. The bridge of claim 15, wherein latency in data transfer from between bus one and bus two is absent after the first bus and second bus are connected to each other.

CLPR:

17. The bridge of claim 15 further comprising a fifth mode of operation, responsive to receiving a selection of the bridge from the second bus after requesting access to the second bus, in which the bridge issues a signal to deassert the selection from second bus.

CLPR:

18. The bridge of claim 15 further comprising a fifth mode of operation, responsive to receiving a selection of the bridge from the second bus after requesting access to the second bus, in which the bridge issues a signal to

deassert the selection from first bus.

CLPR:

19. The bridge of claim 15, wherein the first bus is an advanced system bus.

CLPR:

20. The bridge of claim 15, wherein the first bus is a peripheral component interconnect bus.

CLPR:

23. The bridge of claim 21, wherein the first bus and the second bus are a peripheral component interconnect bus.

CLPR:

25. A method in a data processing system for facilitating a data transfer between a master device and a target device, wherein the data processing system includes a first bus connected to a second bus by a bridge the method comprising:

CLPV:

selecting the bridge in response to determining that the target device is located on the second bus;

CLPV:

initiating a bus request for the second bus by the bridge in response to the selection of the bridge; and

CLPV:

connecting the first bus and the second bus in response to the bridge receiving a grant to the second bus, wherein the master device transfers data between the master device and the target device.

CLPV:

issuing a signal to deselect the bridge in response to the bridge receiving a select form the second bus.

CLPV:

a bridge connecting the first bus to the second bus;

CLPV:

selection means for selecting the bridge in response to determining that the target device is located on the second bus;

CLPV:

initiation means for initiating a bus request for the second bus by the bridge in response to the selection of the bridge; and

CLPV:

connection means for connecting the first bus and the second bus in response to the bridge receiving a grant to the second bus, wherein the master device transfers data between the master device and the target device.

CLPV:

signal means for issuing a signal to deselect the bridge in response to the

bridge receiving a select from the second bus.

CLPV:

a second bridge, connecting the second bus to the third bus;

CLPV:

second selection means for selecting the second bridge in response to determining that the target device is located on the third bus;

CLPV:

second initiation means for initiating a request for the third bus by the second bridge in response to the selection of the second bridge; and

CLPV:

second connection means for connecting the second bus and the third bus in response to the second bridge receiving a grant to the third bus, wherein the master device transfers data between the master device and the target device.

CLPV:

wherein the bridge has a plurality of modes of operation including:

CLPV:

a first mode of operation, responsive to a selection of the bridge originating from the first bus, in which the bridge determines whether a selection also has occurred from the second bus;

CLPV:

a second mode of operation, responsive to a selection of the bridge from both the first bus and the second bus, in which the bridge issues a signal to deassert the selection from the second bus;

CLPV:

a third mode of operation, responsive to a selection of the bridge only from the first bus, in which the bridge issues a request for access to the second bus; and

CLPV:

a fourth mode of operation, responsive to receiving a grant of the second bus, in which the bridge connects the first bus to the second bus.

CLPV:

a bridge connecting the first bus to the second bus;

CLPV:

a decoder connected to the first bus, wherein the decoder receives an address for a target device from the master device, determines whether the target device is connected to the second bus in response to receiving the address for the target device, and selects the bridge in response to determining that the target device is located on the second bus,

CLPV:

wherein the bridge initiates a request for the second bus in response to the selection of the bridge, and connects the first bus and the second bus in response to the bridge receiving a grant to the second bus, wherein the master device transfers data between the master device and the target device.

CLPV:

selecting the bridge in response to determining that the target device is located on the second bus;

CLPV:

determining if the bridge has been selected based on a second request from a device connected to the second bus at a same time as the first request; and

CLPV:

initiating a request for the second bus by the bridge if the first request is selected for completion;

CLPV:

connecting the first bus and the second bus in response to the bridge receiving a grant to the second bus, wherein the master device transfers data between the master device and the target device; and

CLPV:

issuing a signal to the master device to deselect the bridge if the second request is selected for completion.

CLPV:

a bridge connecting the first bus to the second bus;

CLPV:

first selection means for selecting the bridge in response to determining that the target device is located on the second bus;

CLPV:

second determination means for determining if the bridge has been selected based on a second request from a device connected to the second bus at a same time as the first request; and

CLPV:

initiation means for initiating a request for the second bus by the bridge if the first request is selected for completion;

CLPV:

connection means for connecting the first bus and the second bus in response to the bridge receiving a grant to the second bus, wherein the master device transfers data between the master device and the target device; and

CLPV:

signal means for issuing a signal to the master device to deselect the bridge if the second request is selected for completion.

✓

DOCUMENT-IDENTIFIER: US 6230216 B1

TITLE: Method for eliminating dual address cycles in a peripheral component interconnect environment

BSPR:

PCI initiator 110 can be integrated into bus bridge 130, as shown, and bus bridge 130 in turn is used to couple PCI bus 120 to a host bus (not shown). Bus bridge 130 is typically a bi-directional bridge and is made up of numerous components; for simplicity, bus bridge 130 is shown as comprising only PCI initiator 110.

DEPR:

Refer now to FIG. 3, which shows an exemplary PCI bus system implemented in computer system 300 in accordance with a PCI-compliant embodiment of the present invention. The PCI bus system of computer system 300 includes PCI bus 320 coupled to PCI initiator 310. In the present embodiment, PCI initiator 310

is integrated into PCI/host bridge 330. PCI/host bridge 330 is a bi-directional PCI bridge (for simplicity, the elements of a bi-directional bridge other than PCI initiator 310 are not shown). PCI/host bridge 330 is used to couple PCI bus 320 to processor 340 via central processing unit (CPU) bus 345 and to main memory 350 via memory bus 355.

DEPR:

Continuing with reference to FIG. 3, in accordance with the PCI specification, when a computer system (e.g., computer system 300) is first powered on, configuration software stored in main memory (e.g., main memory 350) or in another memory location (not shown) is executed by the CPU (e.g., processor 340). The configuration software, generally referred to as the PCI bus enumerator, scans the PCI bus (e.g., PCI bus 320) to determine what PCI devices

exist on the bus and what configuration requirements those devices have. The configuration spaces (e.g., target configuration spaces 319a-d and initiator configuration space 311) and the configuration registers contained therein are thereby interrogated by processor 340. Processor 340 uses the information from

the configuration registers to configure the PCI bus system. Processor 340 communicates this information to PCI/host bridge 330 in order to instruct the bridge to perform configuration read and write transactions.

DEPR:

Thus, continuing with reference to FIGS. 3 and 4, in accordance with the present embodiment of the present invention, processor 340 executes the configuration software (e.g., the PCI bus enumerator) to scan the PCI bus and access PCI target A 312. The configuration software interrogates configuration

register 440a of PCI target A 312 and reads that the device is a 64-bit target. The configuration software also interrogates other PCI targets on PCI bus 320 and determines their respective ranges. The configuration software provides this information to PCI/host bridge 330, which in the present embodiment registers this information in configuration register 440b of PCI initiator 310 as explained above.

CLPR:

8. The computer system of claim 1 wherein said central processing unit is adapted to interrogate said first configuration register of each of said

plurality of target devices and to communicate said address range of each of said plurality of target devices to said initiator device.

CLPR:

13. The method of claim 9 wherein step b) comprises communicating said address range of each of said plurality of target devices to a configuration register of said initiator device.

CLPR:

15. The method of claim 9 wherein step b) comprises a central processing unit of said computer system interrogating said configuration register of each of said plurality of target devices and communicating said address range of each of said plurality of target devices to said configuration register of said initiator device.

CLPR:

20. The method of claim 17 wherein a central processing unit is used to read said bit in said configuration register of each of said plurality of target devices and communicate a value of said bit to said initiator device.

CLPV:

b) communicating said address range of each of said plurality of target devices to said initiator device; and

DOCUMENT-IDENTIFIER: US 5948094 A

TITLE: Method and apparatus for executing multiple transactions within a single arbitration cycle

DRPR:

FIG. 4 is a block diagram of a computer system including peripheral components and a secondary bridge.

DEPR:

FIG. 2a is a block diagram of a computer system including a host bridge 21 which couples processor 20 to Peripheral Component Interconnect (PCI) bus 26. Host bridge 21 contains timer 22 coupled to PCI arbiter 23. Also coupled to host bridge 21 is system memory 24 which contains a plurality of memory buffers

25. Video capture device 27 is coupled to PCI bus 26, as are other PCI agents 28. Video capture device 27 contains buffers Y, U, and V for storing data.

DEPR:

Video capture device 27 competes with other PCI agents 28 coupled to PCI bus 26

for ownership of the PCI bus. Each agent on PCI bus 26 is coupled to PCI arbiter 23 within host bridge 21. PCI arbiter 23 determines which PCI agent on

PCI bus 26 shall be granted ownership of the PCI bus during an arbitration event. Note that for purposes of this discussion, an arbitration event is one in which the arbiter considers requests from all possible agents, rather than some subset of agents, before granting ownership to the winning agent.

DEPR:

FIG. 4 is a block diagram of a computer system including a processor 40 coupled

to a primary PCI bus 45 through host bridge 41. System memory 42 is additionally coupled to host bridge 41 so that both processor 40 and agents coupled to PCI bus 45 can communicate with system memory 42. Bus agent A 43, bus agent B 44, and bridge 46 are coupled to primary PCI bus 45. Secondary bus

47 is coupled to primary bus 45 through bridge 46. Bus agents C, D, and E, 48,

are coupled to secondary bus 47. For the embodiment shown in FIG. 4, secondary

bus 47 is an Industry Standard Architecture (ISA) bus. However, for an alternate embodiment, secondary bus 47 may be an Extended ISA (EISA) bus or a PCI bus.

DEPR:

For one embodiment, the bus agents 48 on secondary bus 47 each have information

to be communicated to bus agent A 43, or require information from bus agent A 43. Therefore, in this embodiment, bus agent A is the target device.

However,

because of the significant time delay which could be incurred for each of bus agents 48 to independently access and communicate with bus agent A in real time, it is desirable to make use of write and/or read prefetch buffers within bridge 46. In this manner, when the arbiter within host bridge 41 grants a request to bridge 46, the data within these buffers of bridge 46 is

communicated to, or information is read from, bus agent A 43, over the course of multiple transactions within a single arbitration cycle.

DEPR:

From the point of view of primary bus 47, the data from bus agents 48, temporarily stored in memory buffer locations within bridge 46, is fragmented because it can only be dealt with by executing multiple transactions. During an arbitration event, the arbiter within host bridge 41 considers requests from all possible agents, which are, as shown in FIG. 4, bus agents 43, 44, and bridge 46. At the completion of one of the arbitration events initiated by the arbiter, bridge 46 is eventually granted ownership of primary bus 45 for execution of a transaction with bus agent A 43. In accordance with one embodiment of the present invention, a timer coupled to the arbiter within host bridge 41 is programmed to expire in enough time to permit bridge 46 to execute multiple transactions with bus agent A 43 before the arbiter initiates another arbitration event. In this manner, fragmented data associated with each of bus agents 48 can be communicated to or from target bus agent A 43 by bridge 46 within the same arbitration cycle.

DEPR:

For an alternate embodiment, a bridge coupling a primary bus to a secondary bus may require communication with the main memory of a computer system. Implementation of this embodiment is described in reference to the system of FIG. 4 wherein system memory 42 is the target. For one particular embodiment, like video capture device 27 of FIG. 2a, bridge 46 independently buffers fragmented data, such as, for example, data from multiple secondary bus agents, into temporary memory buffers before downloading to system memory. During an arbitration event, the arbiter within host bridge 41 considers requests from all possible agents. At the completion of one of the arbitration events initiated by the arbiter, bridge 46 is eventually granted ownership of primary bus 45 for execution of a transaction with system memory 43. A timer coupled to the arbiter within host bridge 41 is programmed to expire in enough time to permit bridge 46 to execute multiple transactions with system memory 42 before the arbiter initiates another arbitration event. In this manner, fragmented data associated with each of bus agents 48 can be communicated to or from the target system memory 42 by bridge 46 within the same arbitration cycle.

CLPR:

12. The arbitration method of claim 7, wherein the fragmented access bus agent is a bridge coupling the first bus to a second bus, a second bus agent and a third bus agent being coupled to the second bus, the second bus agent communicating with a target device during the first transaction, and the third bus agent communicating with the target device during the second transaction.

CLPR:

22. The computer system of claim 21, wherein the bus is a peripheral component interconnect (PCI) bus and the arbiter and the timer are contained within a bridge coupling the PCI bus to a processor and system memory.

DOCUMENT-IDENTIFIER: US 5928325 A

TITLE: Method of dynamically establishing communication of incoming messages to one or more user devices presently available to an intended recipient

BSPR:

U.S. Pat. Nos. 5,555,376 and 5,493,692 to Theimer et al. generally teach methods to deliver messages to mobile users, e.g., within an office building, on devices, e.g., computer terminals or printers, that the users are in proximity to. To determine where people are in relation to plural communication devices disposed throughout a given geographic location, the use of a separate location facility such as an RF tag badge network (or even GPS receivers) is proposed. When the target device is a display device, the communication system includes functionality, by way of a central agent, which determines which proximal display device provides sufficient privacy for a given transmission. The system determines the communication path from the agent to the user. The disclosed methods are merely an extended form of message call-forwarding to a target device capable of receiving the message in its original transmission. The Theimer patents make possible tracking, for example, an individual's location and forwarding a phone (voice) call message to a phone or mobile unit closest in proximity. The forwarding decision does not involve forwarding a message in one type format (e.g., a voice transmission

directed to a currently non-registered specified unit of an identifiable user) to another communication device which is currently active but which is not designed to recognize voice transmissions received in the format of the original target unit.

DEPR:

The present invention can be more fully described with reference to FIGS. 1 and

2. FIG. 1 illustrates a communication system 10 that includes a central agent (server) 15 coupled to a content transformer 20 and a rules memory device 25. In the illustrative embodiment, the central agent 15 is shown communicating with a plurality of communication networks 30-60. Network 50, as shown, is a conventional cellular phone network typically comprising a central controller switch 31, connecting the cellular phone network 50 to the central agent 15 which may be embodied with a communications computer, the switch 31 typically being a standard cellular Multi-Site Controller (MSC), a home location register

(HLR) database 32 coupled to the switch 31, the HLR typically comprises a computer, and a plurality of transmitter/receiver antennas 33 for communicating to a cellular phone 34 subscribers, over a limited number of communication resources 35.

DEPR:

Messages from the central agent 15 are communicated to the central controller switch 33 which queries a home location register (HLR) server 32 to retrieve the ID of the corresponding cellular phone 34, associated with the recipient identified by the central agent 15, and site location 33 of the cellular phone 34. The central controller switch 31 also responds to inquiries from the central agent 15, as to the availability status of any user devices (cellular phone 34) owned by a central-agent-identified recipient of an incoming message.

Within the cellular network 30, any phone 34 may initiate a communication to

another network (e.g., networks 40-60) (or to another phone on the same cellular network 30) by transmitting a request to the central controller switch

31, and can receive a communication message in a predetermined format type and upon registration with the network 30, by the assignment of a communication resource 35 by the central controller switch 31.

DEPR:

Communication network 40 is a conventional graphics computing devices network including a host controller switch 41, an HLR 42 and a plurality of receiver/transmitter antennas 43 for communicating digital image information, among other types of information (transmit messages) to associated portable graphics terminals 44 over assigned communication resources 45.

DEPR:

Referring to FIG. 1, cellular network 30 and graphics display network 40 are in communication, via central agent 15, with network 50 which is a conventional wireline PSTN (public switched telephone network). PSTN 50 facilitates communication of voice and data transmissions, sourced from regular wireline telephones 55 or the like, to wireless network systems, such networks 30, 40.

DEPR:

The central agent 15 integrates the networks 30-60 by providing the functionality to make possible the exchange of messages between the network systems such that a prospective message recipient who for example carries or has available to him multiple user devices can receive a voice message originally intended for his mobile communication unit (e.g., phone 34) but because the unit is currently unavailable, as determined by an HLR inquiry (e.g. HLR 32), the central agent 15 will automatically transform the voice message into a data signal and communicate it instead to the host controller switch 41. The host controller switch 41 in turn takes the necessary action to alternatively transmit the message for display to the graphics terminal 44.

DOCUMENT-IDENTIFIER: US 5615207 A
TITLE: Side bus to dynamically off load main bus

ABPL:

A data communication system includes an express bus, a plurality of local buses, and a plurality of local/express bridges, each local/express bridge connecting a corresponding local bus to the express bus. A plurality of local/local bridges each connect two corresponding local buses. The plurality of local buses and the plurality of local/local bridges comprise a local path. Also provided is a method of communicating information from a sending communication device to a target communication device, comprising the steps of a) determining if the target communication device is on a local bus corresponding to the sending communication device, b) transferring the information from the sending communication device to the target communication device on the local bus corresponding to the sending communication device if the result of step a) is that the target communication device is on the local bus corresponding to the sending communication device, c) transferring the information from the sending communication device to an express bus if the result of step a) is that the target communication device is not on the local bus corresponding to the sending communication device, d) transferring the information from the express bus to a local bus corresponding to the target communication device, and e) transferring the information from the local bus corresponding to the target communication device to the target communication device.

BSPR:

Alternatively, sending communication devices may overload a target device by sending too much data to the target device. This can occur, for example, when a given sending device is granted too much access to the data bus, and attempts to send a large quantity of data to a single target device in a short period of time. Another example of overloading a target device is when a plurality of devices are granted access to the bus in close time proximity, and all of these devices are attempting to transfer data to the same target device.

BSPR:

It is therefore an object of the invention to provide a data communication system comprising an express bus, a plurality of local buses, and a plurality of local/express bridges, each connecting a corresponding local bus to the express bus.

BSPR:

It is a further object of the invention to provide a plurality of local/local bridges, each connecting two corresponding local buses. The plurality of local buses and the plurality of local/local bridges can comprise a local path.

DEPR:

FIG. 1 shows a bus structure in accordance with the invention. Local buses 105, 107, 109 and 110 are interconnected by local/local bridges 111, 113 and 115. Express bus 117 is connected to local buses 105, 107, 109 and 110 by local/express bridges 119, 121, 123 and 125 respectively.

DEPR:

To carry out the process of sending the information via express bus 117, the information travels from sending communication device 202 over local bus 105 to local/express bridge 119. The information then travels over express bus 117 to local/express bridge 123. The information then travels from local/express bridge 123 to local bus 109, where it passes to target communication device 210.

DEPR:

If the answer to step 505 is that the express bus is not free, then the information is transferred to another local bus, as shown in step 509. This transfer will be through a local/local bridge connecting the local bridge of the sending communication device to another local bridge. In the present example, this transfer will occur via local/local bridge 111 to local bus 107. At step 511, a determination is then made as to whether the target communication device is on the new local bus. If so, as shown in step 513, the information is transferred to the target communication device via the new local bus. In our example the target communication device is on local bus 109. Thus, the result of step 511 is that the target communication device is not on the new local bus and the system returns to step 505 to see if the express bus is free. Alternatively, as a result of a NO answer to the inquiry of step 511, the system can automatically proceed to the express bus, causing the information to wait if the express bus is not free.

DEPR:

As the system proceeds to step 505, the system can maintain a record as to which local buses have previously held the information. This information can be stored, for example, in a series of flags within a portion of the information to be transferred. As a result of a transfer, the transferring bus or local/local bridge can then update this flag information.

DEPR:

Thus, if the express bus is once again busy, a transfer will occur to a local bus which has not yet held the information. In our example, therefore, a subsequent local/local transfer will be via local/local bridge 113 to local bus 109, to which the target communication device is attached. As a result of the express bus being busy for two successive inquiries, therefore, the information transfer occurs along a local path comprising local buses 105-109 and local/local bridges 111 and 113.

DEPR:

As described in this example, the wait time is an absolute time period. Alternatively, the system can maintain wait time statistics, such that if an absolute answer is not available, the system can use an average wait time as the most likely wait time. For example, an external arbiter or processor connected to the local bus or functioning as part of the local/express bridge attached thereto can maintain this data and update this data upon each information transfer. Further, in place of averaging the wait times, other statistical techniques may be employed to obtain a weighted wait time.

DEPR:

As an alternative to the "on ramp" situation described above, the local bus structure can be used in an "off ramp" situation. Here, information may be present on the express bus, but may be unable to exit the express bus onto the local bus corresponding to the target communication device. This situation can present itself if the target communication device is overloaded, the local bus corresponding to the target communication device is overloaded, or the local/express bridge leading to the local bus corresponding to the target communication device is overloaded or malfunctioning.

DEPR:

Using the aforementioned example, presume the information from device 202 was transferred to express bus 117 via local/express bridge 119. Thus, information is on express bus 117 for which a target communication device is communication device 210. In an ideal situation, there would be a recognition that the target communication device is on local bus 109, and the information would thus transfer via local/express bridge 123 to local bus 109 and then to communication device 210.

DEPR:

However, presume communication device 210 is busy, and would like the information from device 202 to be delayed. To accommodate the needs of communication device 210, the information can exit the express bus either via local/express bridge 121 to local bus 107, or via local/express bridge 125 to local bus 110. The information would then pass over a local/local bridge before appearing on local bus 109. As a result, the information is delayed in accordance with the needs of communication device 210.

DEPR:

Alternatively, presume local/express bridge 123 is overloaded or malfunctioning such that an ideal information flow is not possible. Instead of waiting for local/express bridge 123 to be available, the information can take one of the paths described above to either local bus 107 or local bus 110. As a result, although the information is delayed, it still reaches target communication device 210.

CLPR:

2. The data communication system of claim 1, wherein the plurality of local buses and the plurality of local/local bridges comprise a local path.

CLPV:

a plurality of local/express bridges, each connecting a corresponding local bus to the express bus; and

CLPV:

a plurality of local/local bridges, each connecting two corresponding local buses.

DOCUMENT-IDENTIFIER: US 5555383 A

TITLE: Peripheral component interconnect bus system having latency and shadow timers

ABPL:

A PCI system is provided with a shadow register and a shadow timer. When a master device sends an address designating a target device that is connected to another bus, the device's latency value is recorded in the shadow register. While the PCI-PCI bridge arbitrates for the target bus, the master's latency timer increments but the shadow timer will not begin to increment until the PCI-PCI bridge receives a grant# from the target's bus and data transmission begins. Accordingly, the bus arbiter will not de-assert the grant# until the shadow timer has reached the latency value or the master device has released the bus after completing its data transmission. This ensures that the master device will be allocated a time period equal to its latency value to transmit data.

BSPR:

One system which has been developed to enable efficient use of the system bus is the Peripheral Component Interconnect (PCI) architecture. In PCI systems, each device is provided with a latency timer and a predetermined latency value.

An exemplary PCI system is shown in FIG. 1. A more detailed explanation of a known PCI system can be found in, for example, PCI Local Bus Specification, Revision 2.0, Copyright 1992, 1993, PCI Special Interest Group, and in PCI to PCI Bridge Architecture Specification, Revision 1.0, 1994 (original issue), PCI

Special Interest Group, which are incorporated herein by reference.

BSPR:

With reference to FIG. 1, CPU 10 is connected to cache 20 and host bridge 30. The host bridge 30 is connected to the system memory 40 and the system bus 50. Access to system bus 50 is controlled by bus arbiter 60, which may comprise an integral part of the system bus 50. System bus 50 is used to allow communication between various peripheral devices, and between the peripheral devices and the host bridge. For purpose of illustration, four peripheral devices 100, 200, 300, and 400, are shown in FIG. 1; however, those skilled in the art will understand that the number of devices can vary depending on the particular system arrangement.

BSPR:

An exemplary PCI multiple bridge system is shown in FIG. 2, wherein elements similar to those shown in FIG. 1 have the same reference numerals. For the purpose of this example, only four peripheral devices 100, 200, 300, and 400, and two busses 80, and 90, are shown.

BSPR:

In FIG. 2, host bridge 30 is connected to primary bus 80 and secondary bus 90 through the PCI-PCI bridge 70. For the purpose of this example, peripheral devices 100 and 200 are shown to be connected to primary bus 80 and peripheral devices 300 and 400 are shown to be connected to secondary bus 90. It will be appreciated by those skilled in the art, however, that other arrangements are possible.

BSPR:

Mastership of primary bus 80 and secondary bus 90 is controlled by bus arbiters 82 and 92 respectively. The bus arbiters 82 and 92 are illustrated as two respective parts of PCI-PCI bridge 70; however, they can alternatively be implemented, for example, as a single element, or multiple elements constituting respective integral parts of the primary bus 80 and secondary bus 90, as will be apparent to those skilled in the art.

BSPR:

In order for peripheral device 100 to transmit data to peripheral device 300, it first must arbitrate for primary bus 80. Accordingly, the I/O-DMA master 110 sends a request (asserts REQ#) to the bus arbiter 82. When the bus arbiter 82 sends the grant#, peripheral device 100 asserts frame# by sending the proper command and the target's address on the respective bus lines (not shown). PCI-PCI bridge 70 recognizes that the target for the address is connected to secondary bus 90 and, accordingly, keeps the master device 100 in a wait state and arbitrates for the secondary bus 90.

BSPR:

If the latency value L.sub.1 is reached prior to PCI-PCI bus 70 receiving grant# from secondary bus 90, then the I/O-DMA master 110 would have only one cycle to transfer data before it would be required to release the primary bus 80. As a result, peripheral device 100 would be able to transfer data during only one cycle instead of the number of cycles defined by its latency value L.sub.1. Accordingly, if this situation occurs, only a small part of the data from device 100 would be transferred to the target device 300, i.e. only data corresponding to one cycle. In addition, primary bus 80 and PCI-PCI bridge 70 would be wastefully controlled by peripheral device 100 during the time PCI-PCI

bridge 70 arbitrates for secondary bus 90. Since data was transmitted only during one cycle, the wasted period is commensurable with the latency value L.sub.1.

BSPR:

Alternatively, PCI-PCI bridge 70 may receive a grant# from secondary bus 90 before latency timer 120 reaches the latency value L.sub.1 but the remaining time may be insufficient to complete transmission of all the data. Therefore, peripheral devices 100 would transfer data over a period shorter than the number of cycle defined by its latency value L.sub.1. Accordingly, part of the latency value L.sub.1 period would be wastefully allocated to establishing the connection to the target device rather than to data transfer.

BSPR:

Moreover, when a master device that has been allocated the maximum permissible latency is initiating a transaction over the PCI-PCI bridge, part of the latency value is expended on arbitrating for the target's bus. If during the arbitration for the target's bus the latency timer expires, then only one cycle of the maximum permissible latency period would be dedicated to data transmission. Accordingly, in a system where devices are allocated the maximum permissible latency period, each incomplete transaction over the PCI-PCI bridge 70, e.g., when only one data cycle has been used for data transfer, will result in longer wasteful periods.

BSPR:

According to the present invention, a PCI system is provided with a shadow register and a shadow timer. When a master device sends an address designating a target device that is connected to another bus, the device's latency value is recorded in the shadow register. The PCI-PCI bridge would then arbitrate for the target bus. During this arbitration period, the latency timer of the master device is incrementing, but the shadow timer will not begin to increment until the PCI-PCI bridge received a grant# and data transmission began. Accordingly, in the system of the present invention, the bus arbiter will not de-assert the grant# until the shadow timer has reached the latency value or the master device released the bus after completing its data transmission. This ensures that the device will be allocated a time period equal to its latency value to transmit data. That is, even if the device's latency timer reaches the latency value, it will not be required to release the bus since the bus arbiter will not de-assert the grant# before the shadow timer reaches the latency value.

DEPR:

A PCI architecture according to an embodiment of the present invention is shown in FIG. 3, in which elements similar to those of FIG. 2 are designated by similar reference numerals. In FIG. 3, PCI-PCI bridge 70 is provided with bus arbiters 82 and 92, shadow register 84, and shadow timer 86. However, it should be appreciated that other arrangements are possible. For example, the number of bus arbiters may vary. In addition, for simplicity only one shadow timer is shown, however, it is preferable to set the number of shadow timers with respective shadow registers to correspond to the number of peripheral devices.

DEPR:

A particular advantage over the system shown in FIG. 2 is exemplified in the cases where communication is transacted between peripheral devices connected to different system busses. I.e., a two level arbitration. For the purpose of example, the case will be described where peripheral device 100 wishes to transmit data to peripheral device 300. As in the system of FIG. 2, peripheral device 100 first arbitrates for primary bus 80. When bus arbiter 82 sends the grant#, peripheral device 100 asserts frame#, sending the proper command and address on the respective bus lines (not shown), and latency timer 120 begins to increment. PCI-PCI bridge 70 recognizes that the target for the address is connected to secondary bus 90 and, accordingly, records the latency value L.sub.1 of peripheral device 100 in shadow register 84, keeps the master device 100 on a wait state, and arbitrates for the secondary bus 90.

DEPR:

In the device of FIG. 3, bus arbiter 82 may de-assert the grant# only if peripheral device 100 has completed data transmission and released the primary bus 80, or after the shadow timer 86 has reached the latency value L.sub.1. The shadow timer 86, however, does not begin to increment until the PCI-PCI bridge 70 receives grant# from bus arbiter 92 and device 100 begins data transmission. Therefore, during the period when PCI-PCI bridge 70 arbitrates for secondary bus 90, the bus arbiter 82 will not de-assert the grant# (i.e.,

device 100 will not release the primary bus 80 because it has not begun, let alone completed, data transfer, and bus arbiter 82 will not de-assert the grant# because shadow timer has not begun counting, let alone reached the latency value L.sub.1).

DEPR:

When the PCI-PCI bridge 70 receives the grant# from bus arbiter 92, device 100 may begin transmitting the data. Consequently, from this point on, every cycle counted by shadow timer 86 would be a data transfer cycle rather than an idle cycle. Moreover, since shadow timer 86 expires only after it reaches the latency value L.sub.1, which is stored in register 84, peripheral device 100 can efficiently use its latency value L.sub.1 period for data transfer purposes.

DEPR:

One can anticipate that, during the period when device 100 is communicating with device 300, another peripheral device connected to secondary bus 90, for example peripheral device 400, may arbitrate for secondary bus 90. However, if device 100 has not completed its data transfer and the shadow timer 86 has not expired, device 100 will not release the primary bus 80 and, consequently, PCI-PCI bridge 70 will not release the secondary bus 90. Therefore, under such conditions, device 400 may not gain access to secondary bus 90.

DEPR:

On the other hand, if device 400 was required to re-arbitrate after secondary bus 90 has been released, it would have caused a wasteful idle time of secondary bus 90, while arbiter 92 decides which device has priority to receive grant#. In order to substantially eliminate this idle period, in the preferred embodiment, bus arbiter 92 is permitted to de-assert the grant# from PCI-PCI bridge 70 and shift it to another requesting device, such as peripheral device 400. As explained above, PCI-PCI bridge 70 will not release the secondary bus 90 until peripheral device 100 has released the primary bus 80. However, since peripheral device 400 has grant#, it may assert frame# as soon as PCI-PCI bridge 70 releases the secondary bus 90. That is, by completing the arbitration during the time device 100 transmits data, a master device may assert frame# as soon as secondary bus 90 is released.

DEPR:

In the preferred embodiment, elements such as the shadow registers and the shadow timers, are incorporated into the PCI-PCI bridge chip. However, as stated above, other arrangements are possible. For example, the shadow registers and timers may be incorporated in each of the respective system busses. Such an example is shown in FIG. 4, in which elements similar to those of FIG. 3 are designated by similar reference numerals.

DEPR:

As mentioned above, the number of shadow timers may alternatively correspond to the number of peripheral devices. In such a case, the shadow timers may be located in the PCI-PCI bridge, in a respective bus to which the respective device is connected, or in each of the respective peripheral devices. However,

it is preferable that the shadow timers be located in the PCI-PCI bridge 70.

CLPR:

4. The computer system as defined in claim 1, wherein said bridge unit further includes a first bus arbiter circuit for controlling access to said first bus, and a second bus arbiter circuit for controlling access to said second bus.

CLPR:

5. The computer system as defined in claim 1, wherein said bridge unit further includes a register for receiving a latency time value from one of said plurality of first and second peripheral units.

CLPR:

7. The computer system as defined in claim 6, wherein said single shadow timer is disposed in said bridge unit.

CLPR:

12. In a computer system having a host bridge connected to a plurality of system busses, each of the system busses connected to at least one of a plurality of peripheral devices, each of the peripheral devices having a respective latency timer and a respectively assigned latency value, said plurality of system busses connected to a bus bridge for permitting communication among said peripheral devices and between any one of said peripheral devices and the host bridge, a method of controlling communication initiated by one of said peripheral devices connected to a first bus of said system busses and defined as a master device, and one of said peripheral devices connected to a second bus of said system busses and defined as a target device, comprising the steps of:

CLPR:

15. A bus bridge for providing connection between a first data bus and a second data bus, said bus bridge having a shadow timer which begins to increment upon establishment of the connection between said first and second data busses, wherein said bus bridge terminates the connection when said shadow timer reaches a programmed value.

CLPR:

16. The bus bridge of claim 15, further comprising a shadow register for storing said programmed value.

CLPR:

17. The bus bridge of claim 16, further comprising a first peripheral device connected to said first data bus and a bus arbiter responsive to a communication from said first peripheral device to arbitrate for said second data bus, thereby establishing said connection.

CLPR:

19. The computer system as defined in claim 18, wherein said programmed value corresponds to said respective latency value of one of said peripheral devices sending data over said bus bridge.

CLPV:

a bridge unit coupled to said central processing unit;

CLPV:

first and second buses, each of which is coupled to said bridge unit;

CLPV:

d. sending an address of said target device to said bus bridge;

CLPV:

e. sending a secondary grant from said second bus to said bus bridge;

CLPV:

a bus bridge for providing connection between said first data bus and said second data bus, said bus bridge having a shadow timer which begins to increment upon establishment of the connection between said first and second data buses, wherein said bus bridge terminates the connection when said shadow timer reaches a programmed value.

DOCUMENT-IDENTIFIER: US 5528391 A

TITLE: Infrared beam steering system using diffused infrared light and liquid crystal apertures

ABPL:

A system for scanning a room to find the location of target devices, and then locking onto them with stationary directed beams for two-way communication.

In

one embodiment, a base system in each room comprises an IR source/receiver combination plus an LCD display panel which covers the source/receiver and is addressed in such a way as to open up dynamic apertures through which IR radiation in a scanning mode can be directed toward any particular location in the room. When a device at that location senses that it is being irradiated by

the base station, the targeted device responds by emitting a coded packet of IR

pulses. This system takes advantage of the higher bandwidth communication that

can be obtained with point-to-point communications, while still allowing for multiple devices at arbitrary locations in the same room.

BSPR:

One important application of the invention is a system for scanning a room to find the location of target devices, and then locking onto them with stationary

directed beams for two-way communication. A base system in each room comprises

an IR source/receiver combination plus an LCD panel which covers the source/receiver and is addressed in such a way as to open up dynamic apertures through which IR radiation in a scanning mode can be directed toward any particular location in the room. When a device at that location senses that it

is being irradiated by the base station, the targeted device responds by emitting a coded packet of IR pulses. This system takes advantage of the higher bandwidth communication that can be obtained with point-to-point communications, while still allowing for multiple devices at arbitrary locations in the same room.

DEPR:

The system components described are all off-the-shelf components readily available from many suppliers. The addressable LCD display panel 20 can be a conventional active matrix panel with the usual electrical x-y addressing that allows under the control of appropriate signals from the computer 23 a selected

cluster of LCD pixels in the shape of a circle to be switched from their normal

non-transmissive or opaque state to their transmissive state when approximately

3-10% of incident radiation from the source 18 will pass through the aperture 25 in a narrow beam 26 confined by the opaque boundaries of the aperture 25.

For a normal size storeroom, meeting room, or office space, sufficient IR power

exists in the IR rays that can see a particular target device to enable the establishment of the high bandwidth communication link with the device.

DEPR:

The IR receivers both at the device or target end 14-16, and 19 at the source end could have, for example, a high gain phototransistor as the IR detector, and suitable amplifiers to produce a signal to activate an IR source on the device. An example of one simple way to implement the invention is to incorporate an inexpensive 4-bit microcontroller held in reset condition by a signal from a battery source, with the internal amplifiers operating a switch to release the reset condition to cause the microcontroller to execute a simple built-in program that sends a sequence of signals to an IR source on the device to flash it in a predetermined code of long and short flashes equivalent to a UPC bar code. Each device would be programmed with its own unique code pattern. The host computer 23 could easily store in its memory a database comprising the codes for each device and its current location, obtained by periodically activating the system. A simple comparison test of received codes to those stored in the database would allow periodic updating of the database. The above is straightforward programming well within the skills of the average programmer.

DEPR:

FIG. 3 illustrates, in enlarged form, a block diagram of one form of a target device for use in the system of the invention. The device 40 comprises an IR source 41 and IR detector 42 whose output is amplified 43 to operate a switch 44 which via power from a supply 45 normally holds a microcontroller 46 in reset. When the switch is activated, reset is released and the programmed microcontroller 46 generates a sequence of digital signals which amplified 47 can flash the emitter 41 with a built-in code. The microcontroller 46 can then connect 48 with the amplifier to process any received communication signals and be provided with a standard set of responses to be delivered via the emitter 41.

DOCUMENT-IDENTIFIER: US 5239632 A

TITLE: Device to translate logical unit number communications on one SCSI bus to ID communications on a subordinate SCSI bus

BSPR:

As shown, controller 204 includes a microprocessor 206, switching electronics 208, a stored computer program 210, and electronic storage 212 for use by the controller 204 and microprocessor 206. Together, controller 204 routes information received from the SCSI bus 104 to the specific drive 218, 224, 230 as determined by the program stored in controller 204 in program 210. This routing is accomplished using individual buses and control lines for each of the drives. For example, as shown, drive 218 is connected to the controller 204 via a bus 214. In addition, a control line 216 is included, which allows controller 204 to control the operation of drive 218. Similarly, drive 224 includes a bus 220 and a control line 222, and drive 230 includes a bus 226 and a control line 228. It can be seen that this approach requires specific buses and control lines for the controller 204 in order to effect the desired data transfer.

DEPR:

Specifically, the host computer 102 provides a master SCSI ID number and a logical unit number as part of its standard SCSI protocol. The Minnow 304 responds to its master SCSI ID number 6. It maps communications to the devices connected to its subordinate SCSI bus 306 in accordance with the logical unit numbers that are supplied by the host computer 102. Only one master SCSI ID number is used for the master SCSI bus 104. However, through the use of the logical unit members associated with that master SCSI ID number, the Minnow 304 is able to connect up to eight additional devices to the host computer 102 through the use of its mapping function and its subordinate SCSI bus 306. Each of the additional devices connected to its subordinate SCSI bus 306 does not have to be modified since the mapping of the logical unit numbers from the master SCSI bus 104 are converted to subordinate SCSI ID numbers for the subordinate SCSI bus 306 in the manner discussed below. Thus, the Minnow 304 bridges the master SCSI bus 104 with its subordinate SCSI bus 306. This is described below in greater detail.

DEPR:

Referring now to FIG. 4, the master SCSI bus 104 is connected to a master selection machine 402, a master reselection machine 404, and transceivers 406. Master selection machine 402 is connected to an ID switch 406, as is the master reselection machine 404. The ID switch 406, which typically is a DIP or toggle switch of conventional design can be set by the user to specify the master SCSI ID number for the master SCSI bus 104 that the Minnow 304 is set to respond to.

DEPR:

Thereafter, the main control machine 412 via transceiver control signals 418 uses the transceivers 406 to properly transfer the data on the subordinate SCSI

bus 306 to the master SCSI bus 104 and vice versa. In this way, the target and the initiator believe that they are in direct communications with each other. The Minnow 304 can be fabricated using any conventional or future developed approach. It is contemplated that the Minnow 304, with the exception of the ID

switch 406, can readily be implemented in a single chip form utilizing conventional technology. Such a single chip approach is attractive due to the small size, low cost, and low power consumption that would be achieved.

DEPR:

The Minnow 304 remains in the idle state. This changes in a step 708 when the initiator selects a target at the master bus SCSI ID (or address) as set by the

ID switch 406. Thereafter, in a step 710, the master selection machine 402 responds to the selection of the Minnow 304 in accordance with the receipt of the master SCSI ID from the master SCSI bus 104 as set by ID switch 406. In a step 712, the master selection machine 402 handshakes an Identify Message Out received from the initiator on the master SCSI bus 104.

CLPR:

1. A system for allowing more than eight devices to be effectively connected to a master SCSI bus, the system adapted to enable communications to occur between a host device having a first SCSI port, and a plurality of target devices, each having a SCSI port, the system comprising:

CLPR:

7. The system of claim 6, wherein said master reselection machine means comprises an ID switch for permitting the selection of said second master bus SCSI ID number of said minnow means.

CLPR:

9. A system for allowing more than eight devices to be effectively connected to a master SCSI bus, the system adapted to enable communications to occur between a host device having a first SCSI port and a plurality of target devices, each having a SCSI port, the host device connected to the master SCSI bus at the first SCSI, and the plurality of target devices connected to a subordinate SCSI bus, the system transferring communications from the master SCSI bus to a selected target device on the subordinate SCSI bus, the system comprising:

CLPR:

11. The system of claim 10, further comprising an ID switch means to permit selection of the second master bus SCSI ID number to which the master selection machine means responds.

CLPR:

13. The system of claim 12, further comprising an ID switch means to permit the selection of the second master bus SCSI ID number to which the subordinate machine means responds.

CLPR:

14. A method for allowing more than eight units on a master SCSI bus to transfer communications between a host device having a first SCSI port and a selected one of a plurality of target device, each having a SCSI port, the host device connected to the master SCSI bus at the first SCSI port and having a

first master bus SCSI ID number used to identify the host device on the master SCSI bus, the system having a minnow device having a second SCSI port and a third SCSI port, the minnow device connected to the master SCSI bus at the second SCSI port and to a subordinate SCSI bus at the third SCSI port, and the plurality of target devices connected to a subordinate SCSI bus, the method comprising:

CLPV:

(c) minnow means having a second SCSI port and a third SCSI port, for transferring the communications between the host device and one of the plurality of target devices selected by the host device, said minnow means connected to said master SCSI bus at said second SCSI port and to said subordinate SCSI bus at said third SCSI port, said minnow means having a second

master bus SCSI ID number used to identify said minnow means on said master SCSI bus and a first subordinate SCSI ID number used to identify said minnow means on said subordinate SCSI bus, and for converting a SCSI logical unit number received from the host device to a second subordinate bus SCSI ID number, said second subordinate bus SCSI ID number identifying said selected target device on said subordinate SCSI bus to establish communications between the host device and said selected target device.

CLPV:

an ID switch to permit the setting of said second master bus SCSI ID number of said minnow means; and

BSPR:

	L #	Hits	Search Text	DBs
1	L9	41686	5and 6	USPAT; US-PGPUB
2	L8	1	7 and 4	USPAT; US-PGPUB
3	L7	2	5 same 6	USPAT; US-PGPUB
4	L6	41168	(indicat\$4 or identify\$4) adj3 (fail\$4 or trouble\$ or problem\$ or error\$)	USPAT; US-PGPUB
5	L5	389	1 with 2	USPAT; US-PGPUB
6	L4	736343	router\$ or switch\$3 or bridge\$	USPAT; US-PGPUB
7	L3	102287	(plural\$4 or multipl\$4) adj2 (unit\$ or device\$ or system\$)	USPAT; US-PGPUB
8	L2	4216	target adj1 (device\$ or unit\$ or system\$)	USPAT; US-PGPUB
9	L12	19	10 and 4	USPAT; US-PGPUB
10	L11	0	10 and 4 and 6	USPAT; US-PGPUB
11	L10	23	5 same 3	USPAT; US-PGPUB
12	L1	602528	communicat\$4	USPAT; US-PGPUB

DOCUMENT-IDENTIFIER: US 5925120 A

TITLE: Self-contained high speed repeater/lun converter which controls all SCSI operations between the host SCSI bus and local SCSI bus

BSPR:

As shown, controller 204 includes a microprocessor 206, switching electronics 208, a stored computer program 210, and electronic storage 212 for use by the controller 204 and microprocessor 206. Together, controller 204 routes information received from the SCSI bus 104 to the specific drive 218, 224, 230 as determined by the program stored in program 210 of controller 204. This routing is accomplished using individual buses and control lines for each of the drives. For example, as shown, drive 218 is connected to the controller 204 via a bus 214. In addition, a control line 216 is included, which allows controller 204 to control the operation of drive 218. Similarly, drive 224 includes a bus 220 and a control line 222, and drive 230 includes a bus 226 and a control line 228. It can be seen that this approach requires specific buses and control lines for the controller 204 in order to effect a desired data transfer.

BSPR:

U.S. Pat. No. 5,239,632 entitled "DEVICE TO TRANSLATE LOGICAL UNIT NUMBER COMMUNICATIONS ON ONE SCSI BUS TO ID COMMUNICATIONS ON A SUBORDINATE SCSI BUS", incorporated herein by reference, solved many of the problems of the prior art by use of a SCSI LUN converter utilizing state machines and transfer gates to bridge between a main SCSI bus and a subordinate SCSI bus supporting additional devices, as shown in FIGS. 3 and 4.

BSPR:

Specifically, host computer 102 provides a master SCSI ID number and a logical unit number as part of its standard SCSI protocol. The Minnow 304 responds to its master SCSI ID number 6. It maps communications to the devices connected to its subordinate SCSI bus 306 in accordance with the logical unit numbers that are supplied by the host computer 102. Only one master SCSI ID number is used for the master SCSI bus 104. However, through the use of the logical unit numbers associated with that master SCSI ID number, the Minnow 304 is able to connect up to eight additional devices to the host computer 102 through the use of its mapping function and its subordinate SCSI bus 306. None of the additional devices connected to its subordinate SCSI bus 306 have to be modified since the mapping of the logical unit numbers from the master SCSI bus 104 are converted to subordinate SCSI ID numbers for the subordinate SCSI bus 306. Thus, Minnow 304 bridges the master SCSI bus 104 with its subordinate SCSI bus 306.

BSPR:

Referring now to FIG. 4, the master SCSI bus 104 is connected to a master selection machine 402, a master reselection machine 404, and transceivers 406. Master selection machine 402 is connected to an ID switch 406, as is the master reselection machine 404. The ID switch 406 can be set by the user to specify

the master SCSI ID number for the master SCSI bus 104 that the Minnow 304 is set to respond to.

BSPR:

Although this LUN converter solution does meet the specifications for the SCSI bus timing and connects the local target devices "directly" with the master SCSI bus 104 with the use of buffers used as switches to connect and disconnect

signals, it is deficient in several critical areas. First, this solution does not address error conditions on the bus, such as when the host computer 102 tries to use a SCSI message which is not supported by LUN converter. There are

many messages defined in SCSI, not all of which apply to all SCSI devices. Also, it is not mandatory for all SCSI devices to accept all SCSI messages.

If

the host computer 102 uses one of the messages that is not supported with the LUN converter, the LUN converter must be able to give the correct response back

to the host computer 102. This LUN converter solution also does not handle communications from the host computer 102 directed at the LUN converter itself instead of a local target device such as device reset messages without the identify which imply resetting all LUN's within the target.

BSPR:

The controller can be another ASIC with memory or a small microcontroller/microprocessor. It's responsibilities include power on self test, Corona configuration and control, SCSI host bus type selection and monitoring, LUN Select and Reselect processing, SCSI message processing, SCSI error handling. Information needed by the controller can be supplied through any number of standard means such as an EEPROM, a parallel port, switches, or even a jumper bay.

CLPR:

1. A system for allowing more than eight devices to be effectively connected to a single narrow host SCSI bus, the system adapted to enable communications to occur between a host device having a first SCSI port, and a plurality of target devices, each having a SCSI port, the system comprising:

CLPR:

7. A system for allowing more than eight devices to be effectively connected to a single narrow host SCSI bus, the system adapted to enable communications to occur between a host device having a first SCSI port, and a plurality of target devices, each having a SCSI port, the system comprising:

CLPR:

9. A system for allowing more than eight devices to be effectively connected to a single narrow host SCSI bus, the system adapted to enable communications to occur between a host device having a first SCSI port, and a plurality of target devices, each having a SCSI port, the system comprising:

CLPR:

11. A system for allowing more than eight devices to be effectively connected to a single narrow host SCSI bus, the system adapted to enable communications to occur between a host device having a first SCSI port, and a plurality of target devices, each having a SCSI port, the system comprising:

CLPR:

14. A system for allowing more than eight devices to be effectively connected

to a single narrow host SCSI bus, the system adapted to enable communications to occur between a host device having a first SCSI port, and a plurality of target devices, each having a SCSI port, the system comprising:

CLPR:

16. A method for allowing more than eight units on a host SCSI bus to transfer communications between a host device having a first SCSI port and a selected on of a plurality of target devices, each having a SCSI port, the host device connected to the host SCSI bus at the first SCSI port and having a first host bus SCSI ID number used to identify the host device on the host SCSI bus, the system having a Corona device having a second SCSI port and a third SCSI port, the Corona device connected to the host SCSI bus at the second SCSI port and to a local SCSI bus at the third SCSI port, and the plurality of target devices connected to a local SCSI bus, the method comprising:

CLPV:

Corona means having a second SCSI port and a third SCSI port, for transferring the communications between the host device and one of the plurality of target devices selected by the host device, said Corona means connected to said host SCSI bus at said second SCSI port and to said local SCSI bus at said third SCSI port, said Corona means having a second host bus SCSI ID number used to identify said Corona means on said host SCSI bus and a first local SCSI ID number used to identify said Corona means on said local SCSI bus, and for converting a SCSI logical unit number received from the host device to a second local bus SCSI ID number, said second local bus SCSI ID number identifying said selected target device on said local SCSI bus to establish communications between the host device and said selected target device, wherein said Corona means further comprises data path logic, connected between said controller and said host SCSI bus and said local SCSI bus, for converting communications between said host SCSI bus and said local SCSI bus into appropriate data mode; and

CLPV:

Corona means having a second SCSI port and a third SCSI port, for transferring the communications between the host device and one of the plurality of target devices selected by the host device, said Corona means connected to said host SCSI bus at said second SCSI port and to said local SCSI bus at said third SCSI port, said Corona means having a second host bus SCSI ID number used to identify said Corona means on said host SCSI bus and a first local SCSI ID number used to identify said Corona means on said local SCSI bus, and for converting a SCSI logical unit number received from the host device to a second local bus SCSI ID number, said second local bus SCSI ID number identifying said selected target device on said local SCSI bus to establish communications between the host device and said selected target device, wherein said Corona means further comprises arbitration logic, connected to said host SCSI bus at said second SCSI port and to said local SCSI bus at said third SCSI port, for handling arbitration operations between said host SCSI bus and said local SCSI bus, and data path logic, connected between said controller and said host SCSI bus and said local SCSI bus, for converting communications between said host

SCSI bus and said local SCSI bus into the appropriate data mode; and

CLPV:

Corona means having a second SCSI port and a third SCSI port, for transferring the communications between the host device and one of the plurality of target devices selected by the host device, said Corona means connected to said host SCSI bus at said second SCSI port and to said local SCSI bus at said third SCSI

port, said Corona means having a second host bus SCSI ID number used to identify said Corona means on said host SCSI bus and a first local SCSI ID number used to identify said Corona means on said local SCSI bus, and for converting a SCSI logical unit number received from the host device to a second

local bus SCSI ID number, said second local bus SCSI ID number identifying said

selected target device on said local SCSI bus to establish communications between the host device and said selected target device, wherein said second SCSI port of said Corona means has host single-ended and differential SCSI ports to enable said Corona means to be connected to and support either a single-ended or a differential host SCSI bus; and

CLPV:

Corona means having a second SCSI port and a third SCSI port, for transferring the communications between the host device and one of the plurality of target devices selected by the host device, said Corona means connected to said host SCSI bus at said second SCSI port and to said local SCSI bus at said third SCSI

port, said Corona means having a second host bus SCSI ID number used to identify said Corona means on said host SCSI bus and a first local SCSI ID number used to identify said Corona means on said local SCSI bus, and for converting a SCSI logical unit number received from the host device to a second

local bus SCSI ID number, said second local bus SCSI ID number identifying said

selected target device on said local SCSI bus to establish communications between the host device and said selected target device; and

CLPV:

Corona means having a second SCSI port and a third SCSI port, for transferring the communications between the host device and one of the plurality of target devices selected by the host device, said Corona means connected to said host SCSI bus at said second SCSI port and to said local SCSI bus at said third SCSI

port, said Corona means having a second host bus SCSI ID number used to identify said Corona means on said host SCSI bus and a first local SCSI ID number used to identify said Corona means on said local SCSI bus, and for converting a SCSI logical unit number received from the host device to a second

local bus SCSI ID number, said second local bus SCSI ID number identifying said

selected target device on said local SCSI bus to establish communications between the host device and said selected target device; and

✓

Transaction Capabilities Application Part (TCAP) (902). These layers (902-904) together are in conformance with the SS7 protocol. The application layer comprises the Mobile Application Part (MAP) (901). It is the MAP layer (901) that is used to communicate control information, as described above, between switching centers, HLRs, and VLRs. For example, the modify mobile-to-mobile message would be sent within the MAP layer (901).

DEPR:

The present invention a method for establishing and maintaining calls between mobile units in single and multiple switching center configurations such that multiple transcoder format conversions are avoided. This is accomplished by providing control messaging capabilities such that switching centers can determine that a mobile-to-mobile call is in progress. Knowing this, the switching centers can instruct transcoders to allow compressed digital voice to be passed in an essentially transparent manner, thereby avoiding multiple format conversions.

CLPR:

1. In a communication system that comprises a switching center and at least one site controller in communication with the switching center via at least two transcoders, a method for the switching center to establish a call, the method comprising the steps of:

CLPR:

10. In a communication system that comprises a plurality of mobile units capable of wirelessly transmitting and receiving compressed digital voice, a first site controller in wireless communication with the plurality of mobile units, a second site controller in wireless communication with the plurality of mobile units, and a switching center that routes non-compressed digital voice, wherein the first site controller is in communication with the switching center via a first transcoder and the second site controller is in communication with the switching center via a second transcoder, a method for the switching center to establish a call between a first mobile unit of the plurality of mobile units and a target unit via the switching center, the method comprising the steps of:

CLPR:

19. In a communication system that comprises a first switching center in communication with a second switching center, a first site controller in wireless communication with a first set of mobile units, and a second site controller in wireless communication with a second set of mobile units, wherein the first switching center is in communication with a home location register and the home location register is in communication with a visiting location register, wherein the second switching center is in communication with the visiting location register, and wherein the first site controller is in communication with the first switching center via a first transcoder and the second site controller is in communication with the second switching center via a second transcoder, a method for call set-up between a first mobile unit of the first set of mobile units and a second mobile unit of the second set of

X

d 11 bib 1-2

L1 ANSWER 1 OF 2 INPADOC COPYRIGHT 2004 EPO on STN

LEVEL 2

AN 27482755 INPADOC ED 19990928 EW 199938 UP 20000508 UW 200018
TI NETWORK MANAGEMENT EVENT CORRELATION IN ENVIRONMENTS CONTAINING
INOOPERATIVE NETWORK ELEMENTS.
IN WALKER, ANTHONY; PULSIPHER, ERIC A.; SMITH, DARREN D.
INS WALKER ANTHONY; PULSIPHER ERIC A; SMITH DARREN D
INA US; US; US
PA HEWLETT-PACKARD COMPANY
PAS HEWLETT PACKARD CO
PAA US
TL English; French; German
LA English
DT Patent
PIT EPA3 PUBL. OF SEARCH REPORT
PI **EP 909056** **A3 19990922**
DS R: AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
AI EP 1998-108915 A 19980515
PRAI US 1997-947219 A 19971008

L1 ANSWER 2 OF 2 WPIX COPYRIGHT 2004 THOMSON DERWENT on STN

AN 1999-217298 [19] WPIX
DNN N1999-160187
TI Network monitor e.g. for distinguishing between broken and inaccessible
network elements.
DC W01
IN PULSIPHER, E A; SMITH, D D; WALKER, A
PA (HEWP) HEWLETT-PACKARD CO
CYC 27
PI EP 909056 A2 19990414 (199919)* EN 21<--
R: AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT
RO SE SI
JP 11184781 A 19990709 (199938) 20
US 6061723 A 20000509 (200030)
ADT EP 909056 A2 EP 1998-108915 19980515; JP 11184781 A JP 1998-278175
19980930; US 6061723 A US 1997-947219 19971008
PRAI US 1997-947219 19971008

=> dis fam

L1 ANSWER 1 OF 2 INPADOC COPYRIGHT 2004 EPO on STN

PATENT FAMILY INFORMATION

AN 27482755 INPADOC

+-----PRAI-----+	+-----AI-----+
US 1997-947219 A 19971008	EP 1998-108915 A 19980515
	JP 1998-278175 A 19980930
	US 1997-947219 A 19971008
+-----AI-----+	+-----PI-----+
EP 1998-108915 A 19980515	EP 909056 A2 19990414
	EP 909056 A3 19990922
JP 1998-278175 A 19980930	JP 11184781 A2 19990709
US 1997-947219 A 19971008	US 6061723 A 20000509

1 priority, 3 applications, 4 publications

24.31

379/93.29; 379/90.01; 379/114; 455/5.1; 455/5.3; 348/14; 348/17;
348/10;
348/15

L20 ANSWER 5 OF 7 USPATFULL

AN 1999:16830 USPATFULL
TI System, method and article of manufacture for communications utilizing
calling, plans in a hybrid network
IN Elliott, Isaac K., Colorado Springs, CO, United States
Krishnaswamy, Sridhar, Cedar Rapids, IA, United States
PA MCI Communications Corporations, Washington, DC, United States (U.S.
corporation)
PI US 5867495 19990202
AI US 1996-758734 19961118 (8)
DT Utility
LN.CNT 12334
INCL INCLM: 370/352.000
INCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000;
379/144.000
NCL NCLM: 370/352.000
NCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000;
379/144.000
IC [6]
ICM: H04L012-66
ICS: H04L012-28; H04L012-56; H04M015-00
EXF 370/352; 370/383; 370/389; 370/390; 370/392; 370/401; 370/410; 370/408;
379/89; 379/90.01; 379/100.11; 379/114; 379/100.13; 379/93.08;
379/93.07; 379/93.14; 379/93.29; 379/144

L20 ANSWER 6 OF 7 USPATFULL

AN 1999:16829 USPATFULL
TI System, method and article of manufacture with integrated video
conferencing billing in a communication system architecture
IN Krishnaswamy, Sridhar, Cedar Rapids, IA, United States
Elliott, Isaac K., Colorado Springs, CO, United States
Reynolds, Tim E., Iowa City, IA, United States
Forgy, Glen A., Iowa City, IA, United States
Solbrig, Erin M., Cedar Rapids, IA, United States
PA MCI Communication Corporation, Washington, DC, United States (U.S.
corporation)
PI US 5867494 19990202
AI US 1996-752271 19961118 (8)
DT Utility
LN.CNT 16241
INCL INCLM: 370/352.000
INCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000
NCL NCLM: 370/352.000
NCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000
IC [6]
ICM: H04L012-66
ICS: H04L012-28; H04L012-56
EXF 370/352; 370/383; 370/389; 370/390; 370/392; 370/401; 370/458; 370/410;
370/256; 379/67; 379/89; 379/93.07; 379/93.08; 379/93.25; 379/100.11;
379/114; 379/201; 379/207; 379/90.01; 455/436

L20 ANSWER 7 OF 7 USPATFULL

AN 94:71668 USPATFULL
TI Apparatus and method for creation of a user definable video displayed
document showing changes in real time data
IN Risberg, Jeffrey S., 3249 Morris Dr., Palo Alto, CA, United States
94303
Skeen, Marion D., 3826 Magnolia Dr., Palo Alto, CA, United States
94306
PI US 5339392 19940816
AI US 1990-636044 19901228 (7)

RLI Continuation-in-part of Ser. No. US 1990-632551, filed on 21 Dec 1990
which is a continuation-in-part of Ser. No. US 1990-601117, filed on 22
Oct 1990, now patented, Pat. No. US 5257369 which is a
continuation-in-part of Ser. No. US 1989-386584, filed on 27 Jul 1989,
now patented, Pat. No. US 5187787
DT Utility
LN.CNT 7121
INCL INCLM: 395/161.000
INCLS: 395/155.000; 364/408.000
NCL NCLM: 345/333.000
NCLS: 345/334.000; 707/501.000
IC [5]
ICM: G06F015-62
ICS: G06F015-16
EXF 364/144-149; 364/155; 364/161; 364/408; 364/411; 364/412; 364/419;
358/84

09/401,874

(FILE 'HOME' ENTERED AT 15:20:13 ON 20 NOV 2000)

FILE 'USPATFULL' ENTERED AT 15:20:33 ON 20 NOV 2000

L1 670 S DIRECTORY SERVICE#
L2 80809 S (ASSIGN? OR HANDL?) (P) (TASK? OR JOB# OR WORK?)
L3 28 S L1 (P) L2
L4 28 S L3 AND SERVER#
L5 23 S L4 AND CLIENT#
L6 0 S COUNTDOWN CLOCK#
L7 42 S COUNTDOWN CLOCK#
L8 0 S L7 AND L5
L9 0 S L7 AND L3
L10 0 S L1 AND L7
L11 4 S L2 AND L7
L12 45709 S (TREE# OR HIERCHICAL)
L13 53415 S (TREE# OR HIERARCHICAL)
L14 291 S L1 AND L13
L15 0 S L14 AND L7
L16 252 S L14 AND SERVER#
L17 74 S L16 AND (CLOCK# OR TIMER#)
L18 33 S L17 AND L2
L19 24 S L18 AND CLIENT#
L20 7 S L19 AND RESOURCE MANAGEMENT#
L21 10850 S (ASSIGN? OR HANDL? OR DISTRIBUT?) (P) (PLURAL? OR
MULTIPL?) (P) (
L22 103 S L21 AND L1
L23 0 S L22 AND L7
L24 23 S L22 AND CLOCK#
L25 12 S L24 AND ((MULTIP? OR PLURAL?) (3A) (SERVER#))
L26 3 S L25 AND RESOURCE MANAGEMENT#

=> d 125 1-12

L25 ANSWER 1 OF 12 USPATFULL
AN 2000:115204 USPATFULL
TI Systems and methods for matching, selecting, narrowcasting, and/or
classifying based on rights management and/or other information
IN Shear, Victor H., Bethesda, MD, United States
Van Wie, David M., Sunnyvale, CA, United States
Weber, Robert P., Menlo Park, CA, United States
PA InterTrust Technologies Corporation, Santa Clara, CA, United States
(U.S. corporation)
PI US 6112181 20000829
AI US 1997-965185 19971106 (8)
DT Utility
LN.CNT 5857
INCL INCLM: 705/001.000
NCL NCLM: 705/001.000
IC [7]
ICM: G06F017-60
EXF 705/1; 705/10; 705/14; 705/40; 705/400; 707/9; 707/10; 380/4

L25 ANSWER 2 OF 12 USPATFULL
AN 2000:102904 USPATFULL
TI Computer telephone system and method for associating data types with a
color making the data type easily recognizable
IN Bayless, Jeanne A., Plano, TX, United States

Black, William B., McKinney, TX, United States
 Brannick, Gary L., Plano, TX, United States
 Lee, Gene W., Plano, TX, United States
 Lloyd, Lora M., Plano, TX, United States
 Mason, Larry P., Fairview, TX, United States
 Mathis, Amy L., Plano, TX, United States
 Steenbergen, James E., Los Gatos, CA, United States
 Stoldt, Mark R., Allen, TX, United States
 Young, Garrett C., Garland, TX, United States
 Young, Gary C., Dallas, TX, United States
 Fissel, James E., Arlington, TX, United States
 Withers, Robert W., Maryland Heights, MO, United States
 PA Davox Corporation, Westford, MA, United States (U.S. corporation)
 PI US 6100873 20000808
 AI US 1998-56717 19980407 (9)
 RLI Division of Ser. No. US 1997-804233, filed on 21 Feb 1997, now
 patented,
 Pat. No. US 5754636 which is a continuation of Ser. No. US 1994-333058,
 filed on 1 Nov 1994, now abandoned
 DT Utility
 LN.CNT 3444
 INCL INCLM: 345/150.000
 INCLS: 345/153.000
 NCL NCLM: 345/150.000
 NCLS: 345/153.000
 IC [7]
 ICM: G09G005-02
 ICS: G09G005-04
 EXF 345/145; 345/146; 345/150; 345/153; 345/156; 345/157; 345/333; 345/334;
 345/348; 345/350; 345/431

 L25 ANSWER 3 OF 12 USPATFULL
 AN 2000:92889 USPATFULL
 TI Remote communications server system
 IN Hong, Kevin, Spring Lake Park, MN, United States
 Damodar, Bhat V, Bangalore, India
 Narasimhan, Sundararajan, Bangalore, India
 Martenson, Dale W., Mounds View, MN, United States
 Sharma, Raghu, North Oaks, MN, United States
 Davis, Jeffrey P., Ham Lake, MN, United States
 Johnson, Gregory R., New Brighton, MN, United States
 PA Multi-Tech Systems, Inc., Mounds View, MN, United States (U.S.
 corporation)
 PI US 6091737 20000718
 AI US 1997-970644 19971114 (8)
 PRAI US 1996-33201 19961115 (60)
 DT Utility
 LN.CNT 2182
 INCL INCLM: 370/431.000
 INCLS: 370/438.000
 NCL NCLM: 370/431.000
 NCLS: 370/438.000
 IC [7]
 ICM: H04L012-28
 ICS: H04L012-40
 EXF 370/431; 370/432; 370/433; 370/435; 370/437; 370/438; 370/439; 370/448;
 370/401; 375/222; 395/200.47; 395/200.49

 L25 ANSWER 4 OF 12 USPATFULL
 AN 2000:89483 USPATFULL
 TI Automated meter reading system
 IN Kelley, Raymond H., Raleigh, NC, United States
 Carpenter, Richard Christopher, Fuquay-Varina, NC, United States
 Lunney, Robert H., Cary, NC, United States
 Martinez, Maureen, Raleigh, NC, United States

PA ABB Power T&D Company Inc., Raleigh, NC, United States (U.S. corporation)
PI US 6088659 20000711
AI US 1998-82647 19980521 (9)
PRAI US 1997-58659 19970911 (60)
DT Utility
LN.CNT 4678
INCL INCLM: 702/062.000
INCLS: 705/412.000; 709/201.000; 709/203.000; 340/870.020
NCL NCLM: 702/062.000
NCLS: 340/870.020; 705/412.000; 709/201.000; 709/203.000
IC [7]
ICM: G06F015-63
EXF 702/62; 702/61; 702/81-84; 702/122; 702/123; 702/179; 702/182; 702/183;
702/186; 702/187; 702/188; 702/FOR170; 702/FOR171; 702/FOR155;
702/FOR103; 702/FOR104; 702/FOR106; 702/FOR111; 702/FOR112; 364/528.26;
364/528.3; 364/528.31; 364/130-133; 364/138; 364/191; 364/192;
395/200.31; 395/200.54; 395/837; 395/838; 395/200.53; 395/200.33;
395/200.58; 395/200.6; 395/682; 395/200.32; 707/10; 707/11; 707/103;
707/104; 707/2; 705/1; 705/7; 705/8; 705/401; 705/404; 705/405;
705/410;
705/412; 340/870.01; 340/870.02; 340/870.16; 340/870.03; 340/870.13;
340/870.18; 340/870.31; 340/657-664; 379/106.03; 379/102.04;
379/106.01;
379/FOR129; 709/200-203; 709/219; 709/244; 709/300; 700/2; 700/3;
700/9;
700/86; 700/87; 700/286; 700/291; 700/295; 700/296

L25 ANSWER 5 OF 12 USPATFULL

AN 2000:41897 USPATFULL

TI Computer telephone system

IN Bayless, Jeanne A., Plano, TX, United States
Black, William B., McKinney, TX, United States
Brannick, Gary L., Plano, TX, United States
Lee, Gene W., Plano, TX, United States
Lloyd, Lora M., Plano, TX, United States
Mason, Larry P., Fairview, TX, United States
Mathis, Amy L., Plano, TX, United States
Steenbergen, James E., Los Gatos, CA, United States
Stoldt, Mark R., Allen, TX, United States
Young, Garrett C., Garland, TX, United States
Young, Gary C., Dallas, TX, United States
Fissel, James E., Arlington, TX, United States
Withers, Robert W., Maryland Heights, MO, United States

PA Davox Corporation, Westford, MA, United States (U.S. corporation)

PI US 6047054 20000404

AI US 1998-56718 19980407 (9)

RLI Division of Ser. No. US 1997-804233, filed on 21 Feb 1997, now patented,

Pat. No. US 5754636, issued on 19 May 1998 And a continuation of Ser. No. US 1994-333058, filed on 1 Nov 1994, now abandoned

DT Utility

LN.CNT 3564

INCL INCLM: 379/202.000
INCLS: 379/158.000; 379/201.000; 379/204.000; 379/205.000

NCL NCLM: 379/202.000
NCLS: 379/158.000; 379/201.000; 379/204.000; 379/205.000

IC [7]

ICM: H04M003-42

EXF 379/157; 379/158; 379/160; 379/202; 379/204; 379/205; 379/206; 379/201;
379/93.17; 379/93.23; 379/203

L25 ANSWER 6 OF 12 USPATFULL

AN 2000:19240 USPATFULL

TI Computer telephone system

IN Bayless, Jeanne A., Plano, TX, United States
 Black, William B., McKinney, TX, United States
 Brannick, Gary L., Plano, TX, United States
 Lee, Gene W., Plano, TX, United States
 Lloyd, Lora M., Plano, TX, United States
 Mason, Larry P., Fairview, TX, United States
 Mathis, Amy L., Plano, TX, United States
 Steenberg, James E., Los Gatos, CA, United States
 Stoldt, Mark R., Allen, TX, United States
 Young, Garrett C., Garland, TX, United States
 Young, Gary C., Dallas, TX, United States
 Fissel, James E., Arlington, TX, United States
 Withers, Robert W., Maryland Heights, MO, United States
 PA Davox Corporation, Westford, MA, United States (U.S. corporation)
 PI US 6026158 20000215
 AI US 1998-56507 19980407 (9)
 RLI Division of Ser. No. US 1997-804233, filed on 21 Feb 1997, now
 patented,
 Pat. No. US 5754636 which is a continuation of Ser. No. US 1994-333058,
 filed on 1 Nov 1994, now abandoned
 DT Utility
 LN.CNT 3714
 INCL INCLM: 379/355.000
 INCLS: 379/088.030; 379/093.230
 NCL NCLM: 379/355.000
 NCLS: 379/088.030; 379/093.230
 IC [7]
 ICM: H04M015-00
 EXF 379/88.03; 379/93.04; 379/93.21; 379/93.23; 379/93.24; 379/352;
 379/355;
 379/356; 379/354

 L25 ANSWER 7 OF 12 USPATFULL
 AN 1999:161141 USPATFULL
 TI Method for video telephony over a hybrid network
 IN Krishnaswamy, Sridhar, Cedar Rapids, IA, United States
 Elliott, Isaac K., Colorado Springs, CO, United States
 Reynolds, Tim E., Iowa City, IA, United States
 Forgy, Glen A., Iowa City, IA, United States
 Solbrig, Erin M., Cedar Rapids, IA, United States
 PA MCI Communications Corporation, Washington, DC, United States (U.S.
 corporation)
 PI US 5999525 19991207
 AI US 1996-751215 19961118 (8)
 DT Utility
 LN.CNT 20754
 INCL INCLM: 370/352.000
 INCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000
 NCL NCLM: 370/352.000
 NCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000
 IC [6]
 ICM: H04L012-66
 ICS: H04L012-28; H04L012-56
 EXF 370/352; 370/383; 370/389; 370/390; 370/392; 370/401; 370/468; 370/463;
 370/493; 370/410; 379/100.13; 379/93.08; 379/93.07; 379/93.14;
 379/93.29; 379/90.01; 379/114; 455/5.1; 455/6.3; 348/14; 348/17;
 348/10;
 348/15

 L25 ANSWER 8 OF 12 USPATFULL
 AN 1999:152479 USPATFULL
 TI Computer telephone system
 IN Bayless, Jeanne A., Plano, TX, United States
 Black, William B., McKinney, TX, United States
 Brannick, Gary L., Plano, TX, United States

Lee, Gene W., Plano, TX, United States
 Lloyd, Lora M., Plano, TX, United States
 Mason, Larry P., Fairview, TX, United States
 Mathis, Amy L., Plano, TX, United States
 Steenbergen, James E., Los Gatos, CA, United States
 Stoldt, Mark R., Allen, TX, United States
 Young, Garrett C., Garland, TX, United States
 Young, Gary C., Dallas, TX, United States
 Fissel, James E., Arlington, TX, United States
 Withers, Robert W., Maryland Heights, MO, United States
 PA Davox Corporation, Weston, MA, United States (U.S. corporation)
 PI US 5991382 19991123
 AI US 1998-56672 19980407 (9)
 RLI Division of Ser. No. US 1997-804233, filed on 21 Feb 1997, now
 patented,
 Pat. No. US 5754636 which is a continuation of Ser. No. US 1994-333058,
 filed on 1 Nov 1994, now abandoned
 DT Utility
 LN.CNT 3676
 INCL INCLM: 379/136.000
 INCLS: 379/113.000; 379/034.000; 379/267.000; 379/164.000
 NCL NCLM: 379/136.000
 NCLS: 379/034.000; 379/113.000; 379/164.000; 379/267.000
 IC [6]
 ICM: H04M015-00
 ICS: H04M001-24; H04M001-00; H04M003-00
 EXF 379/34; 379/112; 379/113; 379/133; 379/134; 379/135; 379/136; 379/137;
 379/140; 379/265; 379/266; 379/267; 379/309; 379/156; 379/162; 379/157;
 379/164
 L25 ANSWER 9 OF 12 USPATFULL
 AN 1999:81294 USPATFULL
 TI Computer telephone system
 IN Bayless, Jeanne A., Plano, TX, United States
 Black, William B., McKinney, TX, United States
 Brannick, Gary L., Plano, TX, United States
 Lee, Gene W., Plano, TX, United States
 Lloyd, Lora M., Plano, TX, United States
 Mason, Larry P., Fairview, TX, United States
 Mathis, Amy L., Plano, TX, United States
 Steenbergen, James E., Los Gatos, CA, United States
 Stoldt, Mark R., Allen, TX, United States
 Young, Garrett C., Garland, TX, United States
 Young, Gary C., Dallas, TX, United States
 Fissel, James E., Arlington, TX, United States
 Withers, Robert W., Maryland Heights, MO, United States
 PA Davox Corporation, Westford, MA, United States (U.S. corporation)
 PI US 5925101 19990720
 AI US 1998-56569 19980407 (9)
 RLI Division of Ser. No. US 1997-804283, filed on 21 Feb 1997, now
 patented,
 Pat. No. US 5754636 which is a continuation of Ser. No. US 1994-333058,
 filed on 1 Nov 1994, now abandoned
 DT Utility
 LN.CNT 3615
 INCL INCLM: 709/219.000
 NCL NCLM: 709/219.000
 IC [6]
 ICM: G06F017-00
 EXF 345/200.47; 345/200.48; 345/200.49; 379/201; 379/355; 379/216; 379/354
 L25 ANSWER 10 OF 12 USPATFULL
 AN 1999:16830 USPATFULL
 TI System, method and article of manufacture for communications utilizing
 calling, plans in a hybrid network

IN Elliott, Isaac K., Colorado Springs, CO, United States
 Krishnaswamy, Sridhar, Cedar Rapids, IA, United States
 PA MCI Communications Corporations, Washington, DC, United States (U.S.
 corporation)
 PI US 5867495 19990202
 AI US 1996-758734 19961118 (8)
 DT Utility
 LN.CNT 12334
 INCL INCLM: 370/352.000
 INCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000;
 379/144.000
 NCL NCLM: 370/352.000
 NCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000;
 379/144.000
 IC [6]
 ICM: H04L012-66
 ICS: H04L012-28; H04L012-56; H04M015-00
 EXF 370/352; 370/383; 370/389; 370/390; 370/392; 370/401; 370/410; 370/408;
 379/89; 379/90.01; 379/100.11; 379/114; 379/100.13; 379/93.08;
 379/93.07; 379/93.14; 379/93.29; 379/144

 L25 ANSWER 11 OF 12 USPATFULL
 AN 1999:16829 USPATFULL
 TI System, method and article of manufacture with integrated video
 conferencing billing in a communication system architecture
 IN Krishnaswamy, Sridhar, Cedar Rapids, IA, United States
 Elliott, Isaac K., Colorado Springs, CO, United States
 Reynolds, Tim E., Iowa City, IA, United States
 Forgy, Glen A., Iowa City, IA, United States
 Solbrig, Erin M., Cedar Rapids, IA, United States
 PA MCI Communication Corporation, Washington, DC, United States (U.S.
 corporation)
 PI US 5867494 19990202
 AI US 1996-752271 19961118 (8)
 DT Utility
 LN.CNT 16241
 INCL INCLM: 370/352.000
 INCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000
 NCL NCLM: 370/352.000
 NCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000
 IC [6]
 ICM: H04L012-66
 ICS: H04L012-28; H04L012-56
 EXF 370/352; 370/383; 370/389; 370/390; 370/392; 370/401; 370/458; 370/410;
 370/256; 379/67; 379/89; 379/93.07; 379/93.08; 379/93.25; 379/100.11;
 379/114; 379/201; 379/207; 379/90.01; 455/436

 L25 ANSWER 12 OF 12 USPATFULL
 AN 1998:55864 USPATFULL
 TI Computer telephone system
 IN Bayless, Jeanne A., Plano, TX, United States
 Black, William B., McKinney, TX, United States
 Brannick, Gary L., Plano, TX, United States
 Lee, Gene W., Plano, TX, United States
 Lloyd, Lora M., Plano, TX, United States
 Mason, Larry P., Fairview, TX, United States
 Mathis, Amy L., Plano, TX, United States
 Steenbergen, James E., Los Gatos, CA, United States
 Stoldt, Mark R., Allen, TX, United States
 Young, Garrett C., Garland, TX, United States
 Young, Gary C., Dallas, TX, United States
 Fissel, James E., Arlington, TX, United States
 Withers, Robert W., Maryland Heights, MO, United States
 PA Answersoft, Inc., Plano, TX, United States (U.S. corporation)
 PI US 5754636 19980519

AI US 1997-804233 19970221 (8)
RLI Continuation of Ser. No. US 1994-333058, filed on 1 Nov 1994, now
abandoned
DT Utility
LN.CNT 3581
INCL INCLM: 379/142.000
INCLS: 379/127.000; 379/201.000; 379/245.000
NCL NCLM: 379/142.000
NCLS: 379/127.000; 379/201.000; 379/245.000
IC [6]
ICM: H04M015-00
ICS: H04M015-06; H04M003-00; H04M003-42
EXF 379/127; 379/130; 379/131; 379/142; 379/96; 379/97; 379/199; 379/201;
379/245; 379/265; 379/266; 379/309

(FILE 'HOME' ENTERED AT 15:20:13 ON 20 NOV 2000)

FILE 'USPATFULL' ENTERED AT 15:20:33 ON 20 NOV 2000

L1 670 S DIRECTORY SERVICE#
L2 80809 S (ASSIGN? OR HANDL?) (P) (TASK? OR JOB# OR WORK?)
L3 28 S L1 (P) L2
L4 28 S L3 AND SERVER#
L5 23 S L4 AND CLIENT#
L6 0 S COUNTDOWN CLOCK#
L7 42 S COUNTDOWN CLOCK#
L8 0 S L7 AND L5
L9 0 S L7 AND L3
L10 0 S L1 AND L7
L11 4 S L2 AND L7
L12 45709 S (TREE# OR HIERCHICAL)
L13 53415 S (TREE# OR HIERARCHICAL)
L14 291 S L1 AND L13
L15 0 S L14 AND L7
L16 252 S L14 AND SERVER#
L17 74 S L16 AND (CLOCK# OR TIMER#)
L18 33 S L17 AND L2
L19 24 S L18 AND CLIENT#
L20 7 S L19 AND RESOURCE MANAGEMENT#

=> d 1-7

L20 ANSWER 1 OF 7 USPATFULL
AN 2000:103537 USPATFULL
TI Clustered file management for network resources
IN Wolff, James J., Santa Barbara, CA, United States
PA Hewlett-Packard Company, Palo Alto, CA, United States (U.S.
corporation)
PI US 6101508 20000808
AI US 1998-60924 19980415 (9)
RLI Continuation-in-part of Ser. No. US 1997-905307, filed on 1 Aug 1997,
now patented, Pat. No. US 5999930
PRAI US 1998-77146 19980306 (60)
US 1996-23218 19960802 (60)
DT Utility
LN.CNT 4128
INCL INCLM: 707/218.000
INCLS: 707/200.000; 707/001.000; 709/200.000; 709/216.000; 709/223.000;
709/224.000; 709/226.000; 709/239.000; 709/246.000
NCLS: 707/001.000; 707/200.000; 709/200.000; 709/216.000; 709/218.000;
709/223.000; 709/224.000; 709/226.000; 709/239.000; 709/246.000
IC [7]
ICM: G06F017-30
EXF 707/1; 707/200; 709/216; 709/223; 709/224; 709/226; 709/239; 709/246;
709/200; 709/218; 711/202; 714/4; 714/7; 395/200.69; 364/187; 364/188;
370/399; 345/349

L20 ANSWER 2 OF 7 USPATFULL
AN 2000:65768 USPATFULL
TI Resource rebalancing in networked computer systems
IN Wolff, James J., Santa Barbara, CA, United States
PA Hewlett-Packard Company, Palo Alto, CA, United States (U.S.
corporation)

PI US 6067545 20000523
AI US 1998-60857 19980415 (9)
RLI Continuation-in-part of Ser. No. US 1997-905307, filed on 1 Aug 1997
PRAI US 1998-77146 19980306 (60)
US 1996-23218 19960802 (60)
DT Utility
LN.CNT 4335
INCL INCLM: 707/010.000
INCLS: 707/001.000; 709/200.000; 709/216.000; 709/223.000; 709/224.000;
709/226.000; 370/238.000; 370/399.000
NCL NCLM: 707/010.000
NCLS: 370/238.000; 370/399.000; 707/001.000; 709/200.000; 709/216.000;
709/223.000; 709/224.000; 709/226.000
IC [7]
ICM: G06F017-30
EXF 707/1; 707/10; 711/202; 709/216; 709/223; 709/224; 709/226; 709/239;
709/246; 709/200; 714/4; 714/7; 395/200.69; 395/200.32; 364/188;
364/187; 370/399; 370/238; 345/349

L20 ANSWER 3 OF 7 USPATFULL

AN 2000:38919 USPATFULL
TI Distributed I/O store
IN Wolff, James J., Santa Barbara, CA, United States
PA Hewlett-Packard Company, Palo Alto, CA, United States (U.S.
corporation)
PI US 6044367 20000328
AI US 1998-60864 19980415 (9)
RLI Continuation-in-part of Ser. No. US 1997-905307, filed on 1 Aug 1997
PRAI US 1998-77146 19980306 (60)
US 1996-23218 19960802 (60)
DT Utility
LN.CNT 4128
INCL INCLM: 707/002.000
INCLS: 707/001.000
NCL NCLM: 707/002.000
NCLS: 707/001.000
IC [7]
ICM: G06F017-30
EXF 707/1; 707/2; 709/216; 709/223; 709/224; 709/226; 709/239; 709/246;
709/200; 711/202; 714/4; 714/7; 395/200.19; 395/200.32; 364/188;
364/187; 370/399; 345/349

L20 ANSWER 4 OF 7 USPATFULL

AN 1999:161141 USPATFULL
TI Method for video telephony over a hybrid network
IN Krishnaswamy, Sridhar, Cedar Rapids, IA, United States
Elliott, Isaac K., Colorado Springs, CO, United States
Reynolds, Tim E., Iowa City, IA, United States
Forgy, Glen A., Iowa City, IA, United States
Solbrig, Erin M., Cedar Rapids, IA, United States
PA MCI Communications Corporation, Washington, DC, United States (U.S.
corporation)
PI US 5999525 19991207
AI US 1996-751215 19961118 (8)
DT Utility
LN.CNT 20754
INCL INCLM: 370/352.000
INCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000
NCL NCLM: 370/352.000
NCLS: 370/389.000; 370/392.000; 379/090.010; 379/093.070; 379/114.000
IC [6]
ICM: H04L012-66
ICS: H04L012-28; H04L012-56
EXF 370/352; 370/383; 370/389; 370/390; 370/392; 370/401; 370/468; 370/463;
370/493; 370/410; 379/100.13; 379/93.08; 379/93.07; 379/93.14;

L20 ANSWER 3 OF 7 USPATFULL
United States Patent

Patent Number: 6044367
Date of Patent: 28 Mar 2000

Distributed I/O store

Inventor(s): Wolff, James J., Santa Barbara, CA, United States
Assignee: Hewlett-Packard Company, Palo Alto, CA, United States (U.S.
corporation)
Appl. No.: 98-60864
Filed: 15 Apr 1998

Related U.S. Application Data

Continuation-in-part of Ser. No. US 1997-905307, filed on 1 Aug 1997

Priority Data

US 1998-77146 6 Mar 1998 (60)
US 1996-23218 2 Aug 1996 (60)

Int. Cl. G06F017-30
Issue U.S. Cl. 707/002.000; 707/001.000
Current U.S. Cl. 707/002.000; 707/001.000
Field of Search 707/1; 707/2; 709/216; 709/223; 709/224; 709/226;
709/239; 709/246; 709/200; 711/202; 714/4; 714/7;
395/200.19; 395/200.32; 364/188; 364/187; 370/399;
345/349

Reference Cited

PATENT DOCUMENTS

Patent Number	Date	Class	Inventor
US 5283897	Feb 1994	395/650.000	Georgiadis et al.
US 5408663	Apr 1995	395/650.000	Miller
US 5495426	Feb 1996	709/226.000	Waclawsky
US 5504894	Apr 1996	395/650.000	Ferguson et al.
US 5537542	Jul 1996	395/184.010	Eilert et al.
US 5539883	Jul 1996	395/200.110	Allon et al.
US 5628005	May 1997	395/608.000	Hurvig
US 5630129	May 1997	395/675.000	Wheat
US 5668943	Sep 1997	714/007.000	Attanasio
US 5706511	Jan 1998	395/621.000	Tomoda
US 5790789	Aug 1998	395/200.320	Suarez
US 5828569	Oct 1998	364/187.000	Fisher
US 5828847	Oct 1998	395/200.690	Gehr
US 5828876	Oct 1998	707/001.000	Fish
US 5832222	Nov 1998	709/216.000	Dziadosz
US 5889520	Mar 1999	345/349.000	Glaser
US 5893086	Apr 1999	707/001.000	Schmuck

Art Unit - 271

Primary Examiner - Black, Thomas G.

Assistant Examiner - Mizrahi, Diane D.

ABSTRACT

The current invention provides a method for improving throughput to or from a resource by allowing multiple **servers** to concurrently access the resource without affecting the integrity of the resource. Generally, by allowing one **server** to handle the administrative management of a resource, while allowing all **servers**, including the administrative *****server*****, to handle the actual passing of data associated with the I/O request, allows for increased bandwidth between **clients** and the resource. An I/O request to a first **server** node is converted into an access portion and a data transfer portion. The access portion is passed to a corresponding administrative **server** node for the resource. Subsequently, the administrative **server** may issue an access grant to the first **server** node. In response, the first **server** completes the data transfer for the resource.

originates from an aware **client**, control is passed to process 1320. At process 1320, the I/O store and forward buffers are sent back over the network to the **client** in the case of a read I/O. Control is then passed to decision process 1322, where a determination is made if the **server** needs to be load balanced based on the stored CFN records 420D-E, illustrated in FIG. 5A. The determination is made. . . threshold control, two embodiments are possible. Control can be forwarded to process 1328, which sends a generic request to the **client** to redirect its I/O. Alternatively, control can be passed to process 1324, where the load balance monitor controls the load. .

CFN, which can handle the I/O is determined. Control is then forwarded to process 1328, where a request that the **client** redirect I/O to the selected CFN is communicated to the aware **client**. Control is then passed to process 1316, where the resources, which were previously frozen in processes 1218 and 1232 of. . .

DETD . . . subroutine 1178 of FIG. 10E. This is where non-read/write I/O operations are handled. Some operations are handled in the standard **client/server** fashion. Some operations are special or new, such as get/set configuration database process 1352/1354, and come into play during process. . . configuration database is set. Control is then passed to process 1356, where commands to open are managed by the metadata **server**. Control is then passed to process 1358, where commands to close a file are managed by the metadata **server**. Control is then passed to process 1360, where commands to create a file are managed by the metadata **server**. Control is then passed to process 1362, where commands to delete a file are managed by the metadata **server**. Control is then passed to process 1364, where commands to flush any cache data of a file to

commit it to stable storage or flush it to a disk file are managed by the metadata **server**. Control is then passed to process 1366, where commands to lock a file are managed by the metadata **server**. Control is then passed to process 1368, where commands to unlock a file are managed by the metadata **server**. Control is then passed to process 1370, where commands to get attributes of a file are managed by the metadata **server**. Control is then passed to process 1372, where commands to set the attributes of a file are managed by the metadata **server**. Control is then passed to process 1374, where **directory services** are managed by the metadata **server**. Control is then passed to process 1376, where the subroutine is exited.

DETD FIG. 10I illustrates the process flow of an aware **client** 102A-B (see FIGS. 1A, 2B), commencing at start block 1400. Control is passed to process 1402, in which the aware **client** is booted and the modules shown in FIG. 2B are loaded. Control is then passed to process 1404, in which. . . Control is then passed to process 1410, in which the available resources are made available for use by the

aware **client** (see FIG. 6). Control is then passed to decision process 1414. In decision process 1414, the command processing module 192 (see FIG. 2B) determines if the **client** is handling an I/O request. If the command being processed is an I/O request, then control is

passed to process. . . FIG. 2B) is responsible for converting the I/O request for a file system into a path specific request to a node/**server**. The redirector module 184 accesses the **resource management** module 186 (see FIG. 2B) which, in turn, accesses the name driver module 194 to determine the actual path. The. . .

time-out interval has expired, control is passed to process 1426. In another embodiment of the invention, process 1424 could initiate **client** load rebalancing when a **client** detects a delay differential from its normal response time from the **server**.

DETD . . . that determination is negative, then control is passed to process 1448. In process 1448, the command is subject to traditional **client server** processing subsequent to which processing control returns to decision process 1414. If, alternatively, it is determined in decision process 1440, . . .

DETD . . . abstract mapping of system resources and paths to those resources is updated to reflect the new, preferred path from the **client** to the resource(s). Control then returns to decision process 1414 for the processing of the next command.

DETD FIG. 11A is a hardware block diagram of a prior art **client**

in terms of a Microsoft Cluster **Server**.

DETD Cluster, Distributed Database, any Distributed Application):
The important aspect of this work group is that the actual applications,
and the **clients** that use them, exist on the computers that collectively make up the clustered file system. All I/O generated in this. . . non-destructive, secure, law-abiding fashion. STOP-1A specifically refers to an I/O carried out by a CFN that is also the Metadata **Server** for the file system in question. STOP-1B specifically refers to an I/O carried out by a CFN that is not the Metadata **Server** for the file system. STOP-1B1 is the communication from the CFN's Disk Reader to the Metadata Supplier of the CFN that is the Metadata **Server**. STOP-1B2 is the communication from the CFN's Metadata Supplier that is the Metadata **Server** sending the block list to the Disk Reader on the CFN that originated the I/O. STOP-1B3 is the I/O to. . .

DETD STOP Type 2A (1,2): The clustered file system I/O capabilities of a given **client** can take two forms which we shall define as normal **clients** and enabled-**clients**. A normal **client** is one, which has no special awareness of the clustered file system, and hence has absolutely no additional software installed. . . namespace of the network, and thereby decides to attach to a single Clustered File System Node ((.degree. FN) as the **server** for access to that share. In this case, the clustered file system is exposed to the public network as a series of symmetric file system **server** entry-points, each giving the **client** an identical view of the file system. All subsequent I/O from this **client** is carried out by the clustered file system through this single CFN. From the normal **client**'s perspective, this all occurs in the same manner as traditional **client/server** I/O today. Availability is dealt with in the traditional way, by retrying the I/O until successful, or erroring out. An. . . occurs, it may become available at a later time, once restarted. In this respect, availability is the same as traditional **client/server** I/O. However, if the I/O recovery errors out, the **client** or application has the option to manually attach to the clustered file system through another CFN in order to retry. . . accomplished through the symmetry provided by the clustered file system.

This is done manually by distributing a group of normal **clients** among different attach points to the clustered file system, via the different CFNs that publish unique attach points in the namespace viewable by the normal **clients**. Distributed applications are supported in the traditional manner, save for much higher scaling limits, because the clustered file system supports. . . a single view of the file system, no matter where it is viewed from, including the range-locking of files. Normal **clients**, attaching to the clustered file system through different CFN points, will see the exact same file system and hence the. . . applications to scale by using range-locking and/or accessing the same files/file systems to distribute its activities. STOP-2A1 is a normal **client**-generated I/O which occurs on the CFN that is the Metadata **Server** for the file system. STOP-2A2 is a normal **client**-generated I/O which occurs on the CFN that is not the Metadata **Server** for the file system.

DETD STOP Type 2B (1,2): An enabled-**client** is one which has special clustered file system-aware software installed. The enabled-**client** has all the capabilities of a normal **client**, with some important additions. Clustered file system awareness allows availability, scaling, symmetry, single system image, and load-balancing to transparently be extended to the public network. The enabled-**client** now views the exposed clustered file system as a single system image, not a group of symmetric nodes. This is an important abstraction that allows the virtualization of the clustered file system.

The software on the enabled-**client** presents this single system image to the operating system and all **client** applications transact through this virtual interface. File software translates the I/O request to the virtual interface to an actual transaction. . . . which, the original I/O is completed successfully back through the virtual interface. Scaling and load-balancing are accomplished automatically as the enabled-**client** is able to redirect I/O to another cluster node at the request of the clustered file system. Distributed applications function. . . . is achieved by allowing any file system I/O to function identically, regardless of which node initiated it. STOP-2B1 is an enabled-**client** generated I/O which occurs on the CFN that is the Metadata **Server** for the file system. STOP-2B2 is an enabled **client** generated I/O which occurs on the CFN that is not the Metadata **Server** for the file system.

DETD Availability: Availability business can continue when a **server** or component fails. STOP 1 availability is provided in terms of Metadata **server** fail-over and fail-back mechanisms, in order that the I/O can be recovered. STOP 2 availability is provided in terms of symmetry and virtualization through the single system image, allowing manual and transparent **client** I/O recovery.

DETD . . . partly by using a distributed lock manager. This allows an application to grow beyond the capacity of the largest available **server**. Multiple, high-speed paths to the data and range-locks, provided by the distributed lock manager, allow distributed applications to scale. STOP-1. . . .

DETD Symmetry: Metadata **Server** and Hemingway **Client** cache coordinates direct storage subsystem access. STOP-1 and STOP-3 can execute applications on the same storage directly. If those are. . . . and STOP-4 can utilize distributed applications that execute at the source, or services of such applications that execute on a **server**/cluster node in the same way. Everyone sees the same file system and can perform functionally identical I/O from anywhere.

DETD FIGS. 3A-C show the functioning of the **server** node software modules, shown in FIG. 2A, for various implementations of distributed I/O handling shown in FIG. 1B.

DETD FIG. 3A shows the software modules required for the administrative **server** 104B to handle both the administrative and data transfer functions associated with an I/O request (See FIG. 1B I/O request. . . . receipt module 142. The I/O request is tagged with the source identifier indicating the origin of the I/O request, e.g. **client** 100A (see FIG. 1B), and that request and tag are passed to the command processing module 154. The command processing module 154 determines that the I/O request should be passed to the **server** configuration driver 156. The **server** configuration driver uses information obtained from the configuration database 120A-C (see FIGS. 1B, 5B) to determine which, among the plurality of **servers** 104B-106B (see FIG. 1B), is designated as the administrative **server** for the requested file system. In the example shown in FIG. 3A, the

Boolean-based response messages. Finally, the access control table includes a semaphore field 1352. The presence of a semaphore in the semaphore field indicates that one of **clients**, 1154 or 1156, has seized control of the access and volume control tables 1206-1208. A **client** process which has written an identifier, in the semaphore field 1352, can alter the privileges associated with each volume and.

DETD is write enabled. Field 1392B indicates that CD-ROM 1166 (see FIG. 12A) is not write enabled. Field 1394 indicates which **client** currently has write access to a specific volume. Field 1394A indicates that **client** 1154 (see FIG. 12A) currently has write access to RAID storage device 1164. Field 1394B indicates that no **client** has write access to CD-ROM 1166 (see FIG. 12A). Field 1388 indicates which **clients** have mount access privileges for each specific volume. A Boolean True indicates that the **client** can mount the volume. A Boolean False indicates the opposite. Field 1396 indicates, for each **client**, the ability to request a change to its current volume settings. A Boolean False indicates that a **client** is not locked out from making change requests, such as read-only to read-write (or vice versa). A Boolean True indicates that a **client** is locked out from making change requests. Field 1384 is a Boolean True/False, indicating if a **client**, with read only privileges, will be updated when changes, to a specific volume, are made to the volume by other **clients**. Field 1386 is a time stamp, indicating the last time at which a **client** received an updated copy of a file directory 1162 (See FIG. 12A). Field 1382 is a time stamp, indicating the last modification time for a specific volume by any **client**. By comparing the last modification time field 1386 to the volume modification time field 1382, the processes 1214-1216 (see FIG. 12A) can determine when a **client** with auto update privileges is in need of a file directory refresh.

DETD Volume.sub.-- Check.sub.-- **Timer** 14xx

DETD This variable is a **timer** that, when expired, indicates that it is time to check the volume to see if it needs to be refreshed.

DETD SB.sub.-- Process.sub.-- **Timer** 14xx

DETD A **timer**, that when expired, indicates that it is time to check the next file system.

DETD Monitor.sub.-- DB.sub.-- **Timer** 14xx

DETD A **timer** that, when expired, indicates that it is time to check for any pending table requests.

DETD F

server processing the request is also the administrative server for the requested file system. Control passes from the server configuration driver to the shared data lock management module 144. This module is called by the command processing module to. . . load balanced driver 168. This module gathers and periodically reports load balancing utilization statistics, which statistics can be utilized for client load balancing (see FIG. 1A). Control is then passed to the I/O store and forward module 166. The I/O store. . .

DETD FIGS. 3B-C show the complementary relationships associated with distributed I/O between an administrative server and a data transfer server, in accordance with the embodiments shown in FIG. 1B. FIG. 31B shows the software modules associated with the handling of an I/O request by the data transfer server 106B, while FIG. 3C shows the software modules associated with handling the administrative portions of the I/O request, initially received by data transfer server 106B, and handled administratively by administrative server 104B.

DETD . . . 154. On the basis of the source and type of request, the command processing module passes the request to the server config driver which determines it is not the administrative server for the resource I/O request. Command processing module 154 then calls disk reader module 150. The disk reader module 150 determines the administrative server for the volume on which the requested file system resides. Control is then passed to the command receipt module 142 which sends to the administrative server the I/O request. If the I/O is read or write, then the logical I/O is passed to the administrative server for translation to physical sectors on the resource to which the read or write I/O request should be directed. The. . . which accumulates utilization statistics based on I/O requests, and which periodically reports these. These statistics are useful when implementing the client load balancing embodiments and resource rebalancing embodiments of the invention, described and discussed above in connection with FIGS. 1A-C. Control. . .

DETD FIG. 3C shows the software modules associated with the handling by an administrative server 104B of a distributed I/O request passed from a data transfer server 106B (see FIGS. 1B, 3B). Processing begins with the receipt of an I/O request. If it is a read or. . . into storage device ID(s) and physical sector list for the distributed I/O request which is received from the data transfer server by command receipt module 142. The request is tagged with source information by the command receipt module and passed to. . . The command processing module determines, on the basis of I/O type and source, that the request is passed to the server configuration driver 156. The server configuration driver 156 obtains a copy of the current configuration database 120 (see FIG. 1B.) Control is then passed to. . . of a block list, is then passed by the command processing module 154 over the network to the data transfer server 106B.

DETD FIGS. 4A-D show the software modules associated with the handling of I/O by an aware client, the handling of a fail-over and fail-back by an aware client, and the passive and active management of load rebalancing by a client.

DETD . . . which of the software modules, described and discussed above in FIG. 2B, are involved in the processing by an aware client of an I/O request Processing begins with an I/O request generated by application modules 196. That request is passed to the command processing module 192. The command processing module determines whether the requested I/O is destined for a client controlled resource

or an externally controlled resource. For externally controlled resources, the command processing module 192 calls the **resource management** module 186. This module is responsible for managing the information about distinct resources available on the network and the connection. . . . abstract mapping of network namespace resources and combines all available paths for each volume through the plurality of nodes, e.g. **servers** (see FIG. 6). The current path for the resource is returned to **resource management** 186. For external I/O requests, the I/O is sent to the appropriate destination

by

the redirector module 184. This module handles communications between the aware **client** and the network. Data passing to or from the **client**, in response the I/O request, is passed between the network and the application modules 196 via the redirector module 184.

DETD . . . the software modules, described and discussed above in connection with FIG. 2B, is associated with the processing by an aware **client** of a fail-over or fail-back on the network. Fail-over refers to the response, by aware **clients** seeking access to a resource, to the failure of a node, e.g. **server**, designated in the name driver module 194 for accessing that resource. Fail-back deals with the behavior of an aware **client** in response to a recovery of a node, e.g. **server**, on the network from a failed condition. The operation begins, in a manner similar to that described and discussed above. . . . for all external resource, the path to the resource needs to be determined. The request is therefore passed to the **resource management** module 186 and to the name driver module 194 to obtain the path. The command processing module 192 passes the. . . .

DETD FIGS. 4C-D show the software modules on the aware **client** associated with what are defined as passive and active embodiments of **client** load rebalancing, introduced above in FIG. 1A. FIG. 4C discloses a software module associated with passive **client** load balancing, while FIG. 4D shows the software modules associated

with

122B1-B3, while **server** 104C handles only file systems 122A2-3. Several embodiments of the load rebalancing embodiment just discussed will be set forth in. . .

DETD . . . and variations thereof can be practiced individually, or in combination, without departing from the teachings of this invention.

For example, **client** load rebalancing and distributed I/O can be combined. **Client** load rebalancing and resource rebalancing can be combined. Distributed I/O and resource rebalancing can be combined. **Client** load rebalancing, distributed I/O, and resource rebalancing can be combined.

DETD FIG. 2A shows the software modules present on **server** 104 for enabling **client** load balancing, distributed I/O, and resource rebalancing embodiments of the current invention. FIG. 2A shows **server** 104 and memory resource 118. **Server** 104 includes a logical I/O unit 130 and a physical I/O unit 132. The logical I/O unit includes an internal. . . module 148, a command processing module 154, a disk reader module 150, a shared data metadata management module 152, a **server** configuration driver 156, a **resource management** module 158, a logical name driver module 160, and a metadata supplier module 162. The physical I/O unit 132 includes. . .

DETD . . . 140, the command receipt module 142, the shared data lock management module 144, the configuration database replicator module 148, the **resource management** module 158, the **server** configuration driver 156, the shared data metadata management module 152, the metadata supplier module 162, the disk reader module 150, and I/O store and forward 166. The **resource management** module 158 is connected to the resource publisher 146 and to the logical name driver module 160. The metadata supplier. . .

DETD . . . are received and queued up, either from internal I/O module 140, from the private network 112 (from a data transfer **server**), or from a normal or aware **client** on network 108. The I/O is thus tagged with the source type for future decision making.

DETD . . . resources on this node. It is the module that actually interacts with the network in order for normal and aware **clients** to figure out which resources are available on this node. The resource publisher 146 interacts with the **resource management** module 158 and logical name driver module 160 to obtain the actual information that should be published in the network. . .

DETD . . . namespace, and provides a path for the logical name driver module 160 to communicate through command processing module 154, and **server** configuration driver 156, to build said namespace mapping information.

DETD . . . list of the other modules it dispatches commands to include shared data lock manager 144, configuration database replicator module 148, **server** configuration driver 156, **resource management** module 158, shared-data metadata management module 152, and disk reader module 150.

DETD . . . of the configuration database 120 (see FIGS. 5A-D) stored in node memory to other nodes as a result of the **server** configuration driver 156 calling it. It is called when a node first appears on the network. during a fail-over, after. . . node failure, or when a node fails back. It guarantees that every online node has an identical copy of the **server** configuration database. These tables reflect the current state of the **servers**/clustered file

system nodes (CFNs) as a whole and specifically the individual state of each node for which the file system is the administrative **server**

- DETD **SERVER** CONFIGURATION DRIVER 156: This module is responsible for managing the **server** configuration database 120 (see FIGS. 5A-D), responding to requests from a node to get a copy of the current **server** configuration database (FIG. 10H process 1352), sending a command to set the configuration database (FIG. 10H process 1354), rebalancing the. . . a node coming up on the network, first time up or during fail-back and fail-over, and determining who the administrative **server** for a volume is, in response to an I/O by examining the **server** configuration database (see FIG. 10B). Command processing module 154 calls **server** configuration driver 156 to determine whether this CFN is the administrative **server** for the I/O in question.
- DETD . . . (FIG. 10H process 1366, 1368). This module also cooperates in the caching and opportunistic locking mechanisms to efficiently cache administrative **server** block lists and break locks requiring cached file buffers to be committed (FIG. 10H step 1364) to stable storage (see. . .
- DETD . . . module is called by command processing module 154, in the case where an I/O operation is requested, in which the **server** configuration driver 156 has indicated that this node is not the administrative **server** for the file I/O operation in question. The disk reader module 150 determines the administrative **server** for the I/O from the **server** configuration driver 156 and sends the I/O request onto the administrative **server** with a source type request message for translation into a physical I/O block list. Upon failure of the administrative **server**, the disk reader module 150 instructs the **server** configuration database to be rebalanced by calling the **server** configuration driver 156. Upon success, the physical I/O translation table is returned from the administrative **server**'s metadata supplier module 162, at which time the disk reader module 150 forwards the physical I/O onto scheduling module 164. . .
- DETD . . . 1B1, during processing in command receipt module 142. This type of I/O operation is a request received by the administrative **server**'s metadata supplier module 162 from a data transfer **server**'s disk reader module 150. The metadata supplier module 162 translates the logical I/O operation into a physical I/O block list.
- DETD . . . places, depending on the embodiment, including, but not limited to, a usage record in the cluster configuration database, a file **server**, or a load-balance monitor. Further, after each I/O operation, it determines if the current I/O utilization has exceeded the configured. . . load-balance utilization threshold. If so, it conducts a determination depending on the embodiment that results in a message to an aware-**client** to either redirect I/O for a particular resource to a specific node (See FIGS. 7A-B), or to redirect I/O to. . .
- DETD . . . simply gets/delivers the data from/to the memory buffers associated with the internal I/O. In the case of I/O originating from **clients**, temporary memory resources are associated with the I/O, and data is gotten/delivered there. Furthermore, **client** generated I/O requires the I/O store and forward module 166 to retrieve data from the **client** network, and send data to the **client** network, depending on whether the operation is write or read, respectively. After the **client** data is transferred, the temporary memory resources are freed to be used at another time.
- DETD FIG. 2B shows software modules associated with an aware **client** 102A-B which interfaces with the network 108 (see FIG. 1A). The aware

client software modules may reside on a **server**, which implements **client** processes, or a stand-alone unit as shown in FIG. 1A. The aware **client** includes a resource subscriber module 182, a redirector module 184, a **resource management** module 186, a fail-over module 188, a load-balancer module 190, a command processing module 192, a name driver module 194,.

DETD . . . network 108 (see FIG. 1A). The redirector module 184 and the resource subscriber 182 are both connected individually to the **resource management** module 186. The redirector module is also connected to the fail-over module 188 and to the application modules 196. The . . . name driver module 194 and to the command processing module 192. The command processing module 192 is connected to

the **resource management** module 186, load-balancer module 190, and to the application modules 196. The name driver module 194 is also connected to the **resource management** module 186.

DETD . . . 182: This module is responsible for retrieving from the network

the namespace describing the resources available for use by the **clients** on the network. It interacts with **resource management** 186 to respond to a request for retrieval, and to redeliver the resource information.

DETD APPLICATION MODULES 196: This module refers to any application (process)

running on the aware-**client** that generates I/O operations. It calls command processing module 192 to carry out the given I/O operation.

DETD . . . destined for an internally controlled resource or externally controlled resource. If it is not a well-known, internally-controlled resource, it calls **resource management** module 186

L18 ANSWER 1 OF 1 USPATFULL
United States Patent

Patent Number: 5157667
Date of Patent: 20 Oct 1992

Methods and apparatus for performing fault isolation and failure analysis in
link-connected systems

Inventor(s): Carusone, Jr., Anthony, Tucson, AZ, United States
Garrigan, Albert W., Wappingers Falls, NY, United States
Hunsinger, Wayne, Endwell, NY, United States
Moffitt, Gerald T., San Jose, CA, United States
Spencer, David E., Lagrangeville, NY, United States
Taylor, Jordan M., Poughkeepsie, NY, United States
Assignee: International Business Machines Corporation, Armonk, NY, United
States (U.S. corporation)
Appl. No.: 90-516387
Filed: 30 Apr 1990

Int. Cl. G06F011-00
Issue U.S. Cl. 371/029.100; 371/016.500; 340/825.100; 455/008.000
Current U.S. Cl. 714/045.000; 340/825.100; 455/008.000; 714/003.000
Field of Search 371/29.1; 371/16.5; 340/825.1; 370/16; 455/8

Reference Cited

PATENT DOCUMENTS

Patent Number	Date	Class	Inventor
US 3707714	Dec 1972	371/029.100	Plumley
US 4108360	Aug 1978	371/016.500	Beismann et al.
US 4207609	Jun 1980	364/200.000	Luiz et al.
US 4396984	Aug 1983	364/200.000	Videki
US 4438494	Mar 1984	371/029.100	Budde et al.
US 4455605	Jun 1984	364/200.000	Cormier et al.
US 4514846	Apr 1985	371/029.100	Federico
US 4554661	Nov 1985	371/029.100	Bannister
US 4633467	Dec 1986	371/016.500	Abel et al.
US 4644532	Feb 1987	370/094.000	George et al.
US 4745593	May 1988	371/033.000	Stewart
US 4791653	Dec 1988	375/116.000	McFarland et al.
US 4881230	Nov 1989	370/016.000	Clark et al.
US 4884263	Nov 1989	370/016.000	Suzuki
US 4962497	Oct 1990	370/060.100	Ferenc et al.

Art Unit - 233

Primary Examiner - Beausoliel, Robert W.

Assistant Examiner - Hua, Ly

Attorney, Agent or Firm - Scheer, Michael J.; Kinnaman, William A.

27 Claim(s), 4 Drawing Figure(s), 3 Drawing Page(s)

ABSTRACT

The invention relates to methods and apparatus for isolating faults in

link-connected systems utilizing fault reports generated from within the system itself. The reports are transmitted to a central location, preferably during a predetermined time period, and are used to create a single error message identifying the probable nature and location of the fault. A preferred embodiment of the invention does not require either the construction or maintenance of systemwide configuration tables, commonly used performing fault location and analysis. Instead, each unit of a pair of link coupled units, initially or on reconnection, interrogates a link adapter at the other end of the link for an identifier that identifies both the remote unit and the remote link adapter. This "nearest neighbor" information is stored locally at each unit, and is transmitted to the central location when an error is detected. The nearest neighbor information, rather than information from a configuration table, may be used to combine multiple records relating to a fault event, locate the fault and diagnose its cause. Additionally, a preferred embodiment of the invention provides a plurality of reporting paths for each unit in the system, so that the failure of a given unit or link does not prevent error information from being communicated to the central location.

- SUMM . . . analyzing faults in link-connected systems such as, for example, data processing systems arranged as a distributed network of host processors, **switches** and control units coupled by a plurality of communication links. More particularly, the invention relates to methods and apparatus for. . .
- SUMM . . . Pat. No. 4,633,467 require configuration information to be maintained and retrieved in order to implicitly determine which units are in **active** communication paths. These units then become the candidates for the fault location.
- SUMM According to the invention described in the U.S. Pat. No. 4,745,593, a test **packet** is sent through the nodes of a network and a timeout scheme is used to check for an anticipated response.. . .
- SUMM . . . the system described in copending patent application Ser. No. 07/429,267, filed Oct. 30, 1989. Application Ser. No. 07/429,267 describes a **switch** and its protocols for making connections between one input/output channel (of a CPU) and either another input/output channel or a. . .
- SUMM The system described in the incorporated copending application uses **switch** units installed between the CPUs and the CUs to allow connectivity from a single CPU network connection to multiple CUs,. . .
- SUMM When a failure occurs on a link, symptoms occur at both ends of that link and may propagate through the **switch** units and appear at both ends of multiple links. The symptoms of a failure thus appear on both ends of. . .
- SUMM Each **switch** and most CUs have multiple link attachments with paths to CPUs so that when a single path or link fails,. . .
- SUMM . . . the central point are from a single incident. A knowledge of the configuration of all of the CPUs, CUs and **switches** could, as indicated hereinbefore, be kept in a table, but there are difficulties in constructing such a table and dynamically. . .
- SUMM . . . reports may not be enough information to isolate a fault. For example, it would be desirable to be able to **identify** a **unit** that **failed** and is itself unable to issue an error report.
- SUMM According to a preferred embodiment of the invention, each **switch**, CPU and CU in the network (like the network described in the incorporated copending patent application) has an identifier which.
- SUMM Whenever a **switch**, CPU, or CU attached to the CPU/CU interface network is connected to a neighboring unit, it exchanges LAIDs with the.
- SUMM In situations where the failure has been propagated through a **switch**, two links become involved. In this case the two pairs of

failure reports, one pair for each link, are known to be from the same failure since they have the unit identifier of **switch** in common and occur in close time proximity to each other. The method and apparatus contemplated by the invention combine. . .

SUMM . . . occur from the other ends of the links attached to those connections. Each of these multiple reports will contain the **failing unit identifier**. According to the invention, these reports are combined, and since the multiple failure reports indicate a single attached unit, the. . .

SUMM Furthermore, according to the preferred embodiment of the invention, whenever a **switch** or control unit attached to the CPU/CU network detects a failure at one of its link attachments to that network, . . .

DRWD . . . in particular a computer system having a plurality of channels connected to a plurality of control units, through a dynamic **switch**, via a plurality of links.

DRWD . . . 1, except that three processors, each having an associated service processor, are shown coupled to four control units via two **switches**. A set of link attachments (adapters) for these units are depicted along with their corresponding unique link adapter IDs (LAID. . .

DETD The I/O subsystem depicted in FIG. 1 includes a dynamic **switch** 10 having a plurality of ports P, each port P connected to one end of a plurality of links 12-18. One of the links, 18, is connected to a dynamic **switch** control unit 20, and each of the other links 12-17 is connected to either a channel, such as channel A. . .

DETD . . . a point-to-point pair of conductors that may physically interconnect a control unit and a channel, a channel and a dynamic **switch** (such as links 12 and 13), a control unit and a dynamic **switch** (such as links 14-17), or, in some cases, a dynamic **switch** and another dynamic **switch**.

DETD . . . be attached to the I/O interface of that channel or control unit. When a link is attached to a dynamic **switch**, it is said to be attached to a port P on that dynamic **switch**. When the dynamic **switch** makes a connection between two dynamic-**switch** ports, the link attached to one port is considered physically connected to the link attached to the other port, and. . .

DETD The dynamic **switch** 10 provides the capability to physically interconnect any two links that are attached to it. The link attachment point on the dynamic **switch** 10 is the dynamic-**switch** port P. Only two dynamic-**switch** ports P may be interconnected in a single connection, but multiple physical connections may exist simultaneously within the same dynamic **switch**. The dynamic **switch** 10 may be constructed as disclosed in U.S. Pat. Nos. 4,605,928; 4,630,045; and 4,635,250 (the "**switch**" patents), incorporated into the referenced copending patent application.

DETD When a connection is established, two dynamic **switch** ports and their respective point-to-point links are interconnected by a **switch** matrix within the dynamic **switch** 10, as explained in the aforementioned **switch** patents, such that the two links are treated and appear as one continuous link for the duration of the connection. When transmitted frames of information are received by one of two connected **switch** ports P, the frames are normally passed from one port to the other for transmission on the other port's link.

DETD Communications using the **switch** depicted in FIG. 1 are governed by two hierarchical levels of functions and serial-I/O protocols, the link level and the. . . determine the structure, size, and integrity of the frame. Link protocols also provide for making the connection through the dynamic **switch** 10 and for other control functions which are unrelated to this invention. Each channel and each control unit contains a. . .

DETD . . . of a link address to a link-level facility occurs when the link-level facility performs initialization. Every frame sent through the **switch** contains link-level addressing which identifies the source and destination of the frame. Specifically, this addressing information consists of the link addresses of the sending link-level facility (source link address) and receiving link-level facility (destination link address). The **switch** uses this addressing information in order to make a connection from the port receiving the frame to the correct port. . . .

DETD . . . three processors (212, 214 and 216) are shown coupled to four control units (232, 234, 236 and 238), via two **switches** (222 and 224), and a set of link attachments (adapters) for these units, along with their corresponding unique link adapter. . . .

DETD Thus, continuing with the illustrative example involving **switch** unit 222 interconnected with CU 236, LAID pair 222-6 and 236-1 is stored at each end of link 256 (i.e.,

DETD . . . the LAID pair 222-6 and 236-1 being somehow transmitted to a central location (such as service processor 272) from both **switch** unit 222 and CU 236. Clearly, the LAID pair from **switch** unit 222 can be communicated over presumably sound links; however, the LAID pair from CU 236 will need to be. . . .

DETD In situations where the failure has been propagated through a **switch**, two links become involved. Thus, considering a different example, if the failure exists on the path from host processor 214. . . pair for each link, are presumed to be from the same failure since they have the unit identifier of the **switch** (**switch** 224) in common and occur in close time proximity to each other. The method and apparatus contemplated by the invention. . . .

DETD In other situations where, considering yet another example, a unit failure occurs (e.g., **switch** 222), multiple link adapters on the unit will fail and multiple reports will occur from the other ends of the links attached to those connections (from all the units attached to **switch** 222 for this example). Each of these multiple reports will contain the **failing unit identifier**. According to the invention, these reports may be combined at the central location (after being reported to the location) by. . . .

DETD Furthermore, according to the preferred embodiment of the invention, whenever a **switch** or control unit attached to the CPU/CU network detects a failure at one of its link attachments to that network,. . . .

DETD . . . the entry under CPU 212. Also shown as part of entry 501 is the substance of a report received from **switch** 222. The report from **switch** 222 also indicated LOL and the nearest neighbor information transmitted was LAID pair 222-1, 212-1. These two reports had matching. . . .

DETD . . . table was constructed so that the two LOL symptoms result in an analysis that cable 240 (interconnecting CPU 212 and **switch** 222) is faulty, since experience dictates that whenever interconnected units each observe LOL, the interconnecting medium is faulty.

DETD Entry 502 could have similarly been constructed using the nearest neighbor information provided by CPU 212 and **switch** 222. In this case however, the NOS observed by CPU 212 and the LOL observed by **switch** 222 would result in an experience-based diagnosis that the driver associated with port 1 of CPU 212 is faulty or that the receiver associated with port 1 of **switch** 222 is faulty.

DETD . . . expected to report; the depicted state table could be utilized in conjunction with a configuration table rather than with nearest **neighbor** information; **table** entries could be created for a variable number of alternative reporting paths depending on the amount of redundancy one wishes. . . .

L56 ANSWER 1 OF 2 USPATFULL

SUMM Management Information Base (MIB). There are some standard MIB's, such as the IETF (Internet Engineering Task Force) MIB I and MIB II standard definitions. Through the reading and writing of MIB variables, software in other computers can manage or control the component.. . .

DETD (step 418). To perform this check, diagnostic algorithm 400 invokes sink node analyzer algorithm 470 for Node A. If a **problem is identified**, the **Network Monitor** reports that there is a medium probability that node A is causing a TCP problem when operating as a. . . .

DETD ~~For diagnosing network segment problems~~, the above **identified** parameters are also useful with the addition of the alignment rate and the collision rate at the DLL. All or. . . .

L56 ANSWER 2 OF 2 USPATFULL

DETD of the system 60 with a minimal number of measurements or through a minimal number of measurement routes. The measurement **system** 70 also **identifies** or isolates any **problem** module or modules in the system 60 by analyzing the dependencies between the modules and the measurements.

DETD the network interfaces of the various host machines of the ISS 60. These measurements can be made by querying the MIB-II agents that are supported by most host machines. To facilitate more careful planning, the traffic supported by each of the. . . .

=> d 1-2 fp

L56 ANSWER 1 OF 2 USPATFULL

United States Patent

Patent Number: 6115393

Date of Patent: 5 Sep 2000

Network monitoring

Inventor(s): Engel, Ferdinand, Northborough, MA, United States
Jones, Kendall S., Newton Center, MA, United States
Robertson, Kary, Bedford, MA, United States
Thompson, David M., Redmond, WA, United States
White, Gerard, Tyngsborough, MA, United States

Assignee: Concord Communications, Inc., Marlboro, MA, United States (U.S. corporation)

Appl. No.: 95-505083

Filed: 21 Jul 1995

Related U.S. Application Data

Division of Ser. No. US 1991-761269, filed on 17 Sep 1991, now abandoned which is a continuation-in-part of Ser. No. US 1991-684695, filed on 12 Apr 1991, now abandoned

Int. Cl. H04J003-16; H04J003-22

Issue U.S. Cl. 370/469.000

Current U.S. Cl. 370/469.000

Field of Search 370/94.1; 370/85.13; 370/85.14; 370/94.2; 370/110.1;

370/79; 370/241; 370/252; 370/254; 370/465; 370/464;
370/466; 370/467; 370/469; 395/183.15;
395/189.01; 395/189.04; 395/200.04; 395/285; 395/831;
371/20.1

Reference Cited

PATENT DOCUMENTS

Patent Number	Date	Class	Inventor
US 4648061	Mar 1987	340/825.060	Foster
US 4817080	Mar 1989		Soha
US 4887260	Dec 1989		Carden et al.
US 4930159	May 1990		Kravitz et al.
US 5021949	Jun 1991	364/200.000	Morten et al.
US 5025491	Jun 1991		Tsuchiya et al.
US 5038345	Aug 1991	340/825.150	Roth
US 5060228	Oct 1991	370/085.130	Tsutsui et al.
US 5097469	Mar 1992		Douglas
US 5101402	Mar 1992		Chiu et al.
US 5136580	Aug 1992	370/085.130	Videlock et al.
US 5142528	Aug 1992	370/079.000	Kobayashi et al.
US 5142622	Aug 1992	395/200.000	Owens
US 5150464	Sep 1992	395/200.000	Sidhu et al.
US 5347524	Sep 1994	371/020.100	I'Anson et al.
WO 8806822	Sep 1988		

OTHER PUBLICATIONS

Hewlett-Packard brochure regarding local area network protocol analyzer (HP 4972A), Jun. 1987.

F. Kaplan et al., "Application of Expert Systems to Transmission Maintenance", IEEE, 1986, pp. 449-453.

D.M. Chiu et al., "Studying the User and Application Behaviour of a Large Network", Jun. 30, 1988, pp. 1-23.

R. Sudama et al., "The Design of a Realtime DECnet Performance Monitor", Jul. 15, 1988, pp. 1-23.

B.L. Hitson, "Knowledge-Based Monitoring and Control: An Approach to Understanding the Behavior of TCP/IP Network Protocols", ACM, 1988, pp. 210-221.

A.T. Dahbura et al., "Formal Methods for Generating Protocol Conformance Test Sequences", Proceedings of the IEEE, vol. 78, No. 8, Aug. 1990, pp. 1317-1326.

Hewlett-Packard Datasheet brochure, "Analyzing TCP/IP Networks with the HP 4972A", Sep. 1989, pp. 1-8.

Art Unit - 278

Primary Examiner - Patel, Ajit

Attorney, Agent or Firm - Fish & Richardson P.C.

25 Claim(s), 48 Drawing Figure(s), 38 Drawing Page(s)

ABSTRACT

Monitoring is done of communications which occur in a network of nodes, each communication being effected by a transmission of one or more packets among two

51923/P40

Method of reporting errors by a hardware element of a distributed computer system

Inventor(s): Desnoyers, Christine Marie, Pine Bush, NY, United States
 Garmire, Derrick LeRoy, Kingston, NY, United States
 Herrmann, Antoinette Elaine, Poughkeepsie, NY, United States
 Kampf, Francis Alfred, Fairfax, VT, United States
 Stucke, Robert Frederick, Saugerties, NY, United States
 Assignee: International Business Machines Corporation, Armonk, NY, United States (U.S. corporation)
 Appl. No.: 97-831255
 Filed: 8 Apr 1997

Int. Cl. G06F011-00
 Issue U.S. Cl. 395/185.010; 395/182.020; 395/185.100
 Current U.S. Cl. 714/048.000; 714/004.000; 714/057.000
 Field of Search 395/185.01; 395/182.02; 395/183.14; 395/184.01;
 395/183.21; 395/183.19; 395/185.09; 395/185.1

Reference Cited

PATENT DOCUMENTS

Patent Number	Date	Class	Inventor	
US 3242058	Mar 1966	202/176.000	Ganley et al.	
US 4503535	Mar 1985	395/184.010	Budde et al.	
US 4510594	Apr 1985	370/015.000	Johnson, Jr.	
US 4546467	Oct 1985	370/013.000	Yamamoto	
US 4660201	Apr 1987	371/061.000	Nakamura	
US 4769811	Sep 1988	370/060.000	Eckberg, Jr. et al.	
US 4777595	Oct 1988	395/200.660	Strecket et al.	
US 4862461	Aug 1989	371/033.000	Blaner	
US 4970714	Nov 1990	370/017.000	Chen et al.	
US 4993015	Feb 1991	370/016.000	Fite, Jr.	
US 5016243	May 1991	370/016.000	Fite, Jr.	
US 5031211	Jul 1991	379/221.000	Nagai et al.	
US 5117352	May 1992	395/575.000	Falek	
US 5155842	Oct 1992	395/575.000	Rubin	
US 5157667	Oct 1992	395/183.210	Carusone, Jr. et al.	<--
US 5161156	Nov 1992	371/007.000	Baum et al.	
US 5218712	Jun 1993	395/800.000	Cutler et al.	
US 5271000	Dec 1993	370/013.000	Engbersen et al.	
US 5274638	Dec 1993	370/085.600	Michihira et al.	
US 5289460	Feb 1994	370/017.000	Drake, Jr. et al.	
US 5301185	Apr 1994	370/016.100	Cherry	
US 5307354	Apr 1994	395/182.020	Cramer et al.	
US 5487061	Jan 1996	370/013.000	Bray	
US 5546391	Aug 1996	370/060.000	Hochschild et al.	
US 5613069	Mar 1997	395/200.150	Walker	
EP 730355	Apr 1996	H04L001-00		

OTHER PUBLICATIONS

"Generic Alerts for X.25 Packet Layer Control Level Errors," IBM Technical Disclosure Bulletin, vol. 32, No. 9B, pp. 196-198 (Feb. 1990).

"Generic Alerts for X.25 Link Access Protocol Balanced and X.25 Physical Level Errors," IBM Technical Disclosure Bulletin, vol. 32, 9B, pp. 189-191 (Feb. 1990).

Derrick Garmire, "IBM Powerparallel Technology Briefing: Interconnection Technologies for High-Performance Computing (RS/6000 SP)," (Jun. 6, 1996).

Derrick Garmire, "The RS/6000 SP High-Performance Communication Network," (Jun. 6, 1996).

Art Unit - 275

Primary Examiner - Hua, Ly V.

Attorney, Agent or Firm - Gonzales, Esq., Floyd A. Heslin & Rothberg, P.C.

6 Claim(s), 7 Drawing Figure(s), 7 Drawing Page(s)

ABSTRACT

An error message is generated by a hardware element of a distributed computer system, when an error is detected. The error message is then forwarded from the hardware element to one or more designated processing nodes of the distributed computer system. The hardware element includes, for instance, a switch element or a communications adapter adapted to report detected errors.

REP US 5157667 Oct 1992 395/183.210 Carusone, Jr. et al.<--
SUMM . . . the task of monitoring for device failures within the computer system. For example, a heartbeat type protocol is used to **periodically** poll each of the devices in the system to determine if it is still active. If a once active device is. . .